

TUGAS AKHIR - TM141585

PERANCANGAN DAN PEMBUATAN SISTEM KONTROL 2 *AXIS FLEXIBLE FIXTURE* BERBASIS *MICROCONTROLLER*

Havif Alvyan Ikhwanudin
NRP 2115 105 001

Dosen Pembimbing
Ir. Sampurno, MT.

DEPARTEMEN TEKNIK MESIN
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2018



TUGAS AKHIR – TM141585

**PERANCANGAN DAN PEMBUATAN SISTEM
KONTROL 2 AXIS *FLEXIBLE FIXTURE* BERBASIS
*MICROCONTROLLER***

HAVIF ALVYAN IKHWANUDIN
NRP. 2115 105 001

Dosen Pembimbing:
Ir. Sampurno, MT.

PROGRAM SARJANA
DEPARTEMEN TEKNIK MESIN
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2018



FINAL PROJECT – TM141585

**DESIGN AND MANUFACTURE OF CONTROL
SYSTEM 2 AXIS FLEXIBLE FIXTURE BASED ON
MICROCONTROLLER**

HAVIF ALVYAN IKHWANUDIN
NRP. 2115 105 001

Academic Supervisor
Ir. Sampurno, MT.

BACHELOR PROGRAM
DEPARTMENT OF MECHANICAL ENGINEERING
FACULTY OF INDUSTRIAL TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA 2018

PERANCANGAN DAN PEMBUATAN SISTEM KONTROL 2 AXIS FLEXIBLE FIXTURE BERBASIS MICROCONTROLLER

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
pada
Program Studi S-1 Departemen Teknik Mesin
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Oleh :

HAVIF ALVYAN IKHWANUDIN

NRP. 2115 105 001

Disetujui oleh Tim Penguji Tugas Akhir :

1. Ir. Sampurno, MT
NIP. 196504041989031002 (Pembimbing)
2. Ari Kurniawan Saputra, ST, MT
NIP. 198604012015041001 (Penguji I)
3. Dinny Harnany, ST, MT
NIP. 2100201405001 (Penguji II)
4. Ir. Nur Husodo, M.Sc
NIP. 196104211987011001 (Penguji III)



SURABAYA

JANUARI, 2018

PERANCANGAN DAN PEMBUATAN SISTEM KONTROL 2 AXIS FLEXIBLE FIXTURE BERBASIS MICROCONTROLLER

Nama Mahasiswa : Havif Alvyan Ikhwanudin
NRP : 2115 105 001
Jurusan : Teknik Mesin FTI-ITS
Dosen Pembimbing : Ir. Sampurno, MT

Abstrak

Perkembangan teknologi industri terus meningkat, khususnya dalam bidang manufaktur. Semua itu dilakukan karena tuntutan pasar yang menginginkan produk-produk dengan kualitas tinggi dan harga yang terjangkau. Kualitas produk sangat tergantung pada teknologi mesin dan kemampuan operator, sedangkan harga produk tergantung pada waktu produksi yang erat kaitannya dengan efisiensi proses produksi. Untuk menghasilkan produk dengan kepresisian dan ketelitian tinggi dapat dilakukan dengan cara mengembangkan teknologi jig and fixture dari proses permesinan, salah satunya adalah dengan cara rancang bangun flexible fixture yang dapat mempermudah setting up benda kerja. Dalam penelitian ini akan dirancang dan dibuat sistem kontrol flexible fixture yang berbasis microcontroller arduino.

Langkah awal dalam pembuatan sistem kontrol adalah menentukan jenis komponen yang akan digunakan untuk membuat sistem kontrol flexible fixture sampai pada proses merakit, kemudian memprogram microcontroller sehingga flexible fixture dapat bergerak secara otomatis. Langkah selanjutnya adalah menganalisa sistem kontrol dengan menggunakan bantuan software MATLAB, mula-mula menentukan persamaan sistem kontrol dalam time domain berupa state space equation atau frequency domain dalam bentuk overall transfer function, kemudian melakukan analisa kestabilan dengan metode-metode seperti nyquist, routh hurwitz, dan root locus. Selanjutnya

mengevaluasi sistem kontrol flexible fixture dengan memberi input berupa step response, sehingga didapatkan peak time, settling time, rise time, dan persentase overshoot.

Hasil analisa frequency response untuk menentukan kestabilan dengan metode nyquist menghasilkan bahwa grafik frequency response tidak mencakup titik $-1 + j0$ pada sistem kontrol motor DC 1 dan motor DC 2. Pengujian kestabilan dengan metode routh hurwitz menyatakan bahwa pada kolom pertama tidak terjadi perubahan tanda, artinya semua akar bernilai positif pada kedua sistem kontrol. Selanjutnya pengujian dengan metode root locus menghasilkan akar-akar karakteristik terletak di sebelah kiri sumbu imajiner sebesar -1824.8 , $-45.6 + 114.2i$, $-45.6 - 114.2i$ pada sistem kontrol motor DC 1 dan -445.41 , $-115.92 + 71.86i$, $-115.92 - 71.86i$ pada sistem kontrol motor DC 2. Dari ketiga kriteria kestabilan tersebut dapat disimpulkan bahwa sistem kontrol dari flexible fixture dalam keadaan stabil. Untuk evaluasi sistem kontrol ini dengan memberi input step response didapatkan nilai time response berupa peak time 0.0283 s, settling time 0.0877 s, rise time 0.0166 s, persentase overshoot 28.5% pada sistem kontrol motor DC 1 dan didapatkan nilai time response berupa peak time 0.0469 s, settling time 0.0335 s, rise time 0.0204 s, persentase overshoot 0.588% pada sistem kontrol motor DC 2. Dari evaluasi tersebut dapat disimpulkan sistem kontrol ini mempunyai respon waktu yang cukup baik, sehingga flexible fixture dapat bergerak dengan cepat dan sesuai dengan setting point yang diinginkan.

Kata kunci : Flexible Fixture, Microcontroller, Time Response, Frequency Response

**DESIGN AND MANUFACTURE OF CONTROL SYSTEM
2 AXIS FLEXIBLE FIXTURE BASED ON
MICROCONTROLLER**

Name of Student : Havif Alvyhan Ikhwanudin
NRP : 2115 105 001
Department : Teknik Mesin FTI-ITS
Academic Supervisor : Ir. Sampurno, MT

Abstract

The development of industrial technology continues to increase, especially in manufacturing. All that is done because of market demands that want products with high quality and affordable price. Product quality is highly dependent on machine technology and operator capability, while product prices depend on production times closely related to the efficiency of the production process. To produce a product with precision and high accuracy can be done by developing jig and fixture technology from machining process, one of them is by flexible fixture design that can simplify setting up workpiece. In this research will be designed and made flexible fixture control system based on arduino microcontroller.

The first step in making the control system is to determine the type of component that will be used to make flexible fixture control system until assembling process, then program the microcontroller so flexible fixture can move automatically. The next step is to analyze the control system using MATLAB software help, firstly to determine the equation of control system in time domain as state space equation or frequency domain in the form of overall transfer function, then perform stability analysis with methods such as nyquist, routh hurwitz, and root locus. Further evaluate flexible fixture control system by giving input in the form of step response, so get peak time, settling time, rise time, and overshoot percentage.

The result of frequency response analysis to determine the stability with nyquist method resulted that the frequency response graph does not include the point $-1 + j0$ on the DC motor 1 control system and the DC motor 2. Stability testing by routh hurwitz method states that in the first column there is no sign change, meaning all roots are positive on both control systems. Further testing by root locus method produces characteristic roots located on the left of the imaginary axis of -1824.8 , $-45.6 + 114.2i$, $-45.6 - 114.2i$ on DC motor 1 control systems and -445.41 , $-115.92 + 71.86i$, $-115.92 - 71.86i$ on DC motor 2 control system. Of the three stability criteria can be concluded that the control system of flexible fixture in a stable state. To evaluate this control system by giving input step response got time response value in the form of peak time 0.0283 s, settling time 0.0877 s, rise time 0.0166 s, 28.5% overshoot percentage on DC motor 1 control system and got time response value in the form of peak time 0.0469 s, settling time 0.0335 s, rise time 0.0204 s, 0.588% overshoot percentage on DC motor 2 control system. From the evaluation it can be concluded that this control system has good enough time response, so flexible fixture can move quickly and according to the setting point desired.

Key words : Flexible Fixture, Microcontroller, Time Response, Frequency Response

KATA PENGANTAR

Puji syukur saya haturkan kehadirat Allah SWT, hanya karena tuntunan-Nya penulis dapat menyelesaikan tugas akhir ini. Tugas akhir ini disusun untuk memenuhi persyaratan kelulusan pendidikan sarjana S-1 di Departemen Teknik Mesin, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya.

Penyusunan tugas akhir ini dapat terlaksana dengan baik atas bantuan dan kerjasama dari berbagai pihak. Pada kesempatan ini penulis ingin mengucapkan terima kasih kepada:

1. Orang tua penulis, khususnya Bapak Sarjono dan Ibu Khayatun yang senantiasa mendoakan, mendorong, dan menyemangati penulis selama proses penyelesaian tugas akhir ini.
2. Saudara kandung penulis, Maurizka Choirunnisa I dan Miftah Abid Aden A yang selalu mendoakan penulis agar segera menyelesaikan masa studinya.
3. Ir. Sampurno, MT. yang selalu memberikan bimbingan dan arahan dalam penulisan tugas akhir ini.
4. Ari Kurniawan S, ST, MT, Dinny Harnany, ST, MT, dan Ir. Nur Husodo, MSc, selaku dosen penguji yang telah memberikan saran dan kritik kepada penulis tentang tugas akhir ini.
5. Segenap dosen dan karyawan Departemen Teknik Mesin FTI-ITS, atas ilmu yang telah diberikan kepada penulis selama ini.
6. Teman-teman Lintas Jalur 2015 yang senantiasa memberi motivasi, menemani, dan meninggalkan kenangan indah bagi penulis selama masa perkuliahan.
7. Farah Adelia yang selalu mendoakan, memberikan motivasi, dan semangat agar penulis segera menyelesaikan masa studinya.

8. Rekan-rekan “Kontrakan Trocoh”, mas Ambon, Gaplok, Luhur, Aang, Be Cool, Rijal, terimakasih atas segala kesan dan kenangan indahny.
9. Semua pihak yang membantu dalam penyelesaian Tugas Akhir ini.

Dengan segala keterbatasan kemampuan serta pengetahuan penulis, tidak menutup kemungkinan tugas akhir ini jauh dari sempurna. Oleh karena itu, penulis bersedia menerima kritik dan saran dari berbagai pihak untuk penyempurnaan lebih lanjut. Semoga hasil penulisan tugas akhir ini dapat bermanfaat bagi semua pihak. Amiin.

Surabaya, Januari 2018

Penulis

DAFTAR ISI

JUDUL	i
HALAMAN PENGESAHAN	iii
ABSTRAK	iv
<i>ABSTRACT</i>	vi
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Tujuan Perancangan	2
1.4 Manfaat Perancangan	3
1.5 Batasan Masalah	3
1.6 Sistematika Penulisan	3
BAB II DASAR TEORI	5
2.1 Tinjauan Pustaka	5
2.2 <i>Microcontroller</i>	6
2.2.1 Daya (<i>Power</i>)	8
2.2.2 <i>Input</i> dan <i>Output</i>	9
2.3 Motor DC	10
2.3.1 Pengertian Motor DC	10
2.3.2 Prinsip Dasar Cara Kerja	11
2.3.3 Prinsip Arah Putaran Motor	15
2.3.4 <i>Elektromotive Force (EMF)</i> /Gaya Gerak Listrik ..	16
2.3.5 Model Matematis Motor DC	17
2.4 Potensiometer	18
2.4.1 Jenis-Jenis Potensiometer	19
2.4.2 Prinsip Kerja Potensiometer	20
2.5 <i>Driver Motor DC H-Bridge</i>	20
2.6 Analisa Kestabilan	22
2.6.1 Konsep Umum Kestabilan	23
2.6.1.1 Analisa Kestabilan dari Letak <i>Pole-Pole</i>	24
2.6.1.2 Kriteria Stabilitas <i>Hurwitz</i>	27

2.6.1.3 Kriteria Stabilitas <i>Routh-Hurwitz</i>	28
2.6.1.4 Kriteria Stabilitas <i>Nyquist</i>	29
BAB III METODOLOGI PERENCANAAN	35
3.1 Tahap Pengumpulan Data	36
3.2 Tahap Perencanaan Otomatisasi <i>Flexible Fixture</i>	36
3.3 Analisa Sistem Kontrol <i>Flexible Fixture</i> dengan <i>Microcontroller</i>	39
3.4 Analisa Kestabilan Sistem Kontrol <i>Flexible Fixture</i>	39
3.5 Pemrograman Sistem Kontrol <i>Flexible Fixture</i>	41
BAB IV PEMILIHAN KOMPONEN DAN ANALISA SISTEM KONTROL <i>FLEXIBLE FIXTURE</i>	43
4.1 Perencanaan Sistem Kontrol <i>Flexible Fixture</i> Berbasis <i>Microcontroller</i>	43
4.1.1 <i>Power Supply</i>	43
4.1.2 <i>Microcontroller</i>	45
4.1.3 Motor DC <i>Driver</i>	45
4.1.4 Motor DC	46
4.1.4.1 Spesifikasi Motor DC 1	46
4.1.4.2 Spesifikasi Motor DC 2.....	47
4.1.5 <i>Multi-turn Potentiometer</i>	47
4.2 Analisa Kecermatan <i>Multi-turn Potentiometer</i>	48
4.3 Analisa Pergerakan <i>Flexible Fixture</i> Berbasis <i>Microcontroller</i>	48
4.3.1 Analisa Pergerakan Miring <i>Flexible Fixture</i>	48
4.3.2 Analisa Pergerakan Putar <i>Flexible Fixture</i>	51
4.4 Pembahasan Sistem Kontrol <i>Flexible Fixture</i>	56
BAB V WIRING DIAGRAM DAN PEMROGRAMAN SISTEM KONTROL <i>FLEXIBLE FIXTURE</i>	59
5.1 Wiring Diagram Sistem Kontrol <i>Flexible Fixture</i>	59
5.2 Pemrograman Sistem Kontrol <i>Flexible Fixture</i> Pada <i>Microcontroller</i>	59
5.3 <i>Sketch</i> Program Sistem Kontrol <i>Flexible Fixture</i>	65
BAB VI ANALISA KESTABILAN SISTEM KONTROL....	67
6.1 Model Matematis Sistem Kontrol	67
6.2 Fungsi Transfer	72

6.3 Analisa Kestabilan dengan Metode <i>Nyquist</i>	74
6.4 Analisa Kestabilan dengan Metode <i>Routh Hurwitz</i>	75
6.5 Analisa Kestabilan dengan Metode <i>Root Locus</i>	77
6.6 Evaluasi Pergerakan Miring dan Putar Sistem Kontrol <i>Flexible Fixture</i>	79
6.6.1 Karakteristik Respon	80
BAB VII KESIMPULAN DAN SARAN	83
7.1 Kesimpulan	83
7.2 Saran	84
DAFTAR PUSTAKA	
LAMPIRAN-LAMPIRAN	
BIODATA PENULIS	

DAFTAR GAMBAR

Gambar 1. 1 <i>Road Map Flexible Fixture</i>	2
Gambar 2. 1 Board Arduino UNO	7
Gambar 2. 2 Motor DC Sederhana	11
Gambar 2. 3 Medan Magnet yang Membawa Arus Mengelilingi Konduktor	12
Gambar 2. 4 Medan Magnet yang Membawa Arus Mengelilingi Konduktor U	12
Gambar 2. 5 Reaksi Garis Fluks	13
Gambar 2. 6 Prinsip Kerja Motor DC	14
Gambar 2. 7 <i>EMF Counter</i>	16
Gambar 2. 8 Diagram Skematik Motor DC.....	17
Gambar 2. 9 Struktur Potensiometer beserta Bentuk dan Simbolnya.....	19
Gambar 2. 10 Jenis-Jenis Potensiometer	20
Gambar 2. 11 Rangkaian <i>Driver Motor DC H-Bridge</i> Transistor.....	21
Gambar 2. 12 Letak <i>Pole</i> dan <i>Zero</i> dalam Bidang s.....	23
Gambar 2. 13 <i>Pole</i> dengan τk Negatif.....	25
Gambar 2. 14 <i>Pole</i> dengan τk Positif	26
Gambar 2. 15 Sistem <i>Loop</i> Tertutup	29
Gambar 2. 16 Pemetaan Bidang Kompleks.....	30
Gambar 2. 17 Jalan Pelingkupan Dua <i>Pole</i>	32
Gambar 2. 18 Diagram <i>Nyquist</i>	32
Gambar 2. 19 Contoh Diagrams <i>Nyquist</i>	34
Gambar 3. 1 Diagram Alir Perencanaan Sistem Kontrol <i>Flexible Fixture</i>	37
Gambar 3. 2 <i>Flexible Fixture</i> yang Direncanakan	37
Gambar 3. 3 Rangkaian Sistem Kontrol yang Direncanakan..	38
Gambar 3. 4 Blok Diagram Sistem Kontrol <i>Flexible Fixture</i> ..	38
Gambar 3. 5 Diagram Simulasi MATLAB	40

Gambar 3. 6 Pemrograman Sistem Kontrol <i>Flexible Fixture</i> Pada <i>Microcontroller</i>	42
Gambar 4. 1 <i>Switching Power Supply</i>	44
Gambar 4. 2 <i>Microcontroller</i> Arduino MEGA	45
Gambar 4. 3 <i>Motor DC Driver</i>	46
Gambar 4. 4 Motor DC	46
Gambar 4. 5 <i>Multi-Turn Potentiometer</i>	47
Gambar 4. 6 Pergerakan Miring <i>Flexible Fixture</i>	49
Gambar 4. 7 Pergerakan Putar <i>Flexible Fixture</i>	51
Gambar 5. 1 Tampilan Arduino <i>Software (IDE)</i>	59
Gambar 5. 2 Tampilan Pemilihan <i>Board</i> Arduino Mega Pada Arduino <i>Software (IDE)</i>	61
Gambar 5. 3 Tampilan Pengecekan <i>Port</i> Arduino <i>Software</i> (<i>IDE</i>) dengan <i>Microcontroller</i>	62
Gambar 5. 4 Tampilan <i>Sketch Editor</i> Pada Arduino <i>Software (IDE)</i>	63
Gambar 5. 5 Tampilan Menu <i>Verify</i> Pada Arduino <i>Software (IDE)</i>	64
Gambar 5. 6 Tampilan Menu <i>Upload</i> Pada Arduino <i>Software (IDE)</i>	65
Gambar 6. 1 Model Matematis Atau Fungsi Alih <i>Microcontroller</i>	67
Gambar 6. 2 Model Matematis Atau Fungsi Alih Motor <i>DC Driver</i>	68
Gambar 6. 3 Diagram Skematik Motor DC	68
Gambar 6. 4 (a) Model Matematis atau Fungsi Alih Motor DC 1, (b) Model Matematis atau Fungsi Alih Motor DC 2	70
Gambar 6. 5 (a) Model Matematis atau Fungsi Alih <i>Gear</i> Motor DC 1, (b) Model Matematis atau Fungsi Alih <i>Gear</i> Motor DC 2	71
Gambar 6. 6 Model Matematis atau Fungsi Alih Sensor	72
Gambar 6. 7 (a) Diagram <i>Block</i> Sistem Kontrol <i>Flexible</i>	

<i>Fixture</i> Pada Motor DC 1, (b) Diagram <i>Block</i> Sistem Kontrol <i>Flexible Fixture</i> Pada Motor DC 2	73
Gambar 6. 8 (a) <i>Nyquist</i> Diagram Sistem Kontrol <i>Flexible Fixture</i> Pada Motor DC 1. (b) <i>Nyquist</i> Diagram Sistem Kontrol <i>Flexible Fixture</i> Motor DC 2	75
Gambar 6. 9 (a) <i>Root Locus</i> Sistem Kontrol <i>Flexible Fixture</i> Pada Motor DC 1, (b) <i>Root Locus</i> Sistem Kontrol <i>Flexible Fixture</i> Pada Motor DC 2	78
Gambar 6. 10 (a) Hasil Grafik <i>Step Response</i> Sistem Kontrol <i>Flexible Fixture</i> Pada Motor DC1, (b) Hasil Grafik <i>Step Response</i> Sistem Kontrol <i>Flexible Fixture</i> Pada Motor DC 2	80
Gambar 6. 11 (a) Karakteristik Respon Sistem Kontrol <i>Flexible Fixture</i> Motor DC 1, (b) Karakteristik Respon Sistem Kontrol <i>Flexible Fixture</i> Motor DC 2	82

DAFTAR TABEL

Tabel 2. 1 Deskripsi Arduino MEGA	7
Tabel 4. 1 Tabel Analisa Kesalahan dan Kepresisian Pergerakan Miring <i>Flexible Fixture</i>	50
Tabel 4. 2 Tabel Analisa Kesalahan Dan Kepresisian Pergerakan Putar <i>Flexible Fixture</i>	52

BAB I

PENDAHULUAN

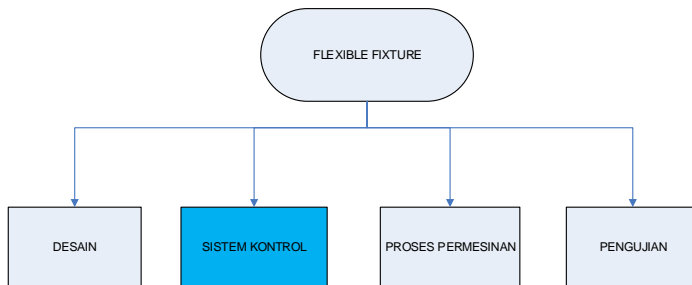
1.1 Latar Belakang

Perkembangan industri yang pesat memberi kesempatan yang baik untuk mengembangkan pemesinan yang menunjang proses produksi. Peran industri memang sangat besar dalam sektor pemesinan karena bisa merangsang perkembangan teknologi dalam suatu mesin untuk dapat memenuhi permintaan dari konsumen. Hal ini disebabkan karena teknologi pemesinan mengandung keterkaitan dengan berbagai industri yang pada dasarnya mencakup segala aspek teknologi, bahan, perancangan dan proses.

Salah satu alat yang dapat digunakan dalam sebuah proses produksi yang bergerak dalam dalam bidang manufaktur adalah *flexible fixture*. Alat ini biasanya digunakan pada mesin milling (*milling machine*), mesin bor (*drilling machine*), mesin milling CNC (*milling CNC machine*), mesin EDM (*Electrical Discharge Machine*), dan lain-lain. *Flexible fixture* dapat berputar dari 0^0 - 360^0 serta bisa bergerak miring mulai 0^0 - 90^0 ke kanan atau ke kiri pada semua posisi putar dari 0^0 - 360^0 . Dengan memakai alat tersebut diharapkan dapat membantu proses produksi menjadi lebih cepat dan dapat menghemat waktu. Penggunaan *microcontroller* sebagai alat untuk mengatur gerakan dari *flexible fixture* dapat membantu dan mempermudah pengerjaan, operator mesin tidak perlu melakukan pengaturan secara manual karena pengaturan gerakan alat diatur dengan cara memasukkan program dari *potentiometer* sesuai dengan putaran dan sudut kemiringan yang diinginkan. Dengan adanya sistem kontrol pada *flexible fixture* diharapkan proses pengerjaan benda kerja dapat akurat dan presisi serta bisa mempercepat dan mempermudah pengerjaan benda kerja.

Apabila penggunaan alat ini mampu diterapkan dalam setiap proses industri manufaktur dalam negeri, maka proses

produksi akan menjadi lebih cepat dan membuat biaya produksi menjadi lebih rendah. Jika biaya produksi menjadi lebih rendah perusahaan akan mendapatkan keuntungan yang lebih besar dan dapat membuat industri manufaktur menjadi lebih berkembang. Gambar 1.1 dibawah merupakan *road map* dalam perancangan *flexible fixture*.



Gambar 1. 1 *Road Map Flexible Fixture*

1.2 Perumusan Masalah

Dalam perencanaan ini perumusan masalah yang digunakan adalah sebagai berikut:

- Bagaimana merancang dan membuat otomatisasi *flexible fixture* berbasis *microcontroller* yang sederhana.
- Bagaimana menganalisa kestabilan sistem kontrol *flexible fixture* berbasis *microcontroller* agar berjalan dengan baik.

1.3 Tujuan Perancangan

Tujuan dalam perencanaan ini adalah sebagai berikut:

- Merancang dan membuat otomatisasi *flexible fixture* berbasis *microcontroller* yang sederhana dan murah.
- Menganalisa kestabilan sistem kontrol *flexible fixture* berbasis *microcontroller* agar berjalan dengan baik dan menghasilkan produk yang sesuai dengan permintaan.

1.4 Manfaat Perancangan

Manfaat yang didapatkan dari perencanaan ini adalah sebagai berikut:

- a. Dapat memberikan kemudahan dan ketelitian dalam proses pemesinan sehingga mendapatkan hasil produk yang baik.
- b. Memberikan alternatif pada industri untuk pemilihan penggunaan mesin selain mesin yang di *import*.
- c. Sebagai studi perbandingan dikalangan industri manufaktur.
- d. Pengaturan dari *microcontroller* pada *flexible fixture* membuat mesin akan lebih efisien digunakan.

1.5 Batasan Masalah

Batasan masalah yang digunakan dalam perencanaan ini adalah sebagai berikut:

- a. Perancangan ini hanya membuat *flexible fixture* dapat bergerak secara otomatis.
- b. Konstruksi *flexible fixture* dianggap ideal.
- c. Spesifikasi *flexible fixture* dengan resolusi 1^0 dan kepresisian $25'$
- d. *Microcontroller* yang digunakan adalah arduino MEGA.
- e. Kestabilan sistem kontrol dicari dengan menggunakan *software* MATLAB.
- f. Motor DC yang digunakan mempunyai spesifikasi yang berbeda dalam pemodelan.

1.6 Sistematika Penulisan

- a. BAB I Pendahuluan

Menjelaskan tentang latar belakang dari tinjauan secara umum, perencanaan, perumusan masalah, tujuan perencanaan, manfaat perencanaan, batasan-batasan masalah serta sistematika penulisan

b. BAB II Tinjauan Pustaka

Berisi tentang uraian tinjauan pustaka yang mendukung penulisan serta sebagai acuan yang dapat digunakan dalam pengerjaan *flexible fixture*.

c. BAB III Metodologi Perencanaan

Berisi tentang uraian langkah-langkah yang dilakukam dalam perencanaan dan analisa sistem kontrol *flexible fixture*.

d. BAB IV Pemilihan Komponen dan Analisa Sistem Kontrol pada *Flexible Fixture*

Bab ini membahas tentang cara kerja, komponen yang mendukung sistem kontrol pada *flexible fixture* dan menganalisa hasil pergerakan *motor DC*.

e. BAB V Wiring Diagram dan Pemrograman Sistem Kontrol *Flexible Fixture*

Merupakan bab yang berisi tentang wiring diagram dan langkah-langkah untuk memprogram *flexible fixture* pada *microcontroller*.

f. BAB VI Analisa Kestabilan Sistem Kontrol

Dalam bab ini berisi tentang analisa kestabilan sistem kontrol pada *flexible fixture* dengan menggunakan *software MATLAB*

g. BAB VII Kesimpulan dan Saran

Bab ini berisi tentang kesimpulan dan saran yang didapatkan dari hasil perencanaan dan analisa kestabilan sistem kontrol *flexible fixture*.

BAB II DASAR TEORI

2.1 Tinjauan Pustaka

Perencanaan proses (*planning process*) dilakukan untuk menerjemahkan informasi rancangan produk kedalam tahapan manufaktur. Pengembangan *Computer Aided Process Planning* (CAPP) dimaksudkan untuk menjembatani kesenjangan yang terjadi antara *Computer Aided Design* (CAD) dengan *Computer Aided Manufacturing* (CAM). Perancangan *fixture* merupakan salah satu tahapan yang harus dilakukan dalam perencanaan. Namun sebagian besar penelitian CAPP mengabaikan pengaruh perancangan *fixture* terhadap rancangan produk dan perencanaan proses. Padahal *fixture* memegang peranan yang sangat penting untuk melakukan proses pemesinan.

Ala P (2011) melakukan penelitian tentang perancangan sistem kontrol *flexible fixture* berbasis *microcontroller* yang menggunakan aktuator motor *stepper* [5]. Hasil yang didapat yaitu presentase *error* kemiringan *flexible fixture* dengan nilai *error* terendah sebesar 0%, dan nilai *error* tertinggi sebesar 10%, dengan rata-rata nilai *error* sebesar 4.62%. Penyebab *error* tersebut disebabkan karena resolusi motor *stepper* sebesar 0.9^0 pada setiap stepnya. Pratama N, dkk (2014) melakukan penelitian tentang analisa kontrol posisi *motor DC* berbasis algoritma ANFIS menggunakan aplikasi arduino [6]. Hasil yang didapatkan yaitu nilai *error* terbaik dari pengujian dengan sudut referensi 270 derajat (dengan beban) sebesar 0.7%. *Error* yang dihasilkan *motor DC* lebih kecil dibandingkan *error* yang dihasilkan dengan motor *stepper*, sehingga pada perancangan *flexible fixture* kali ini menggunakan *motor DC* yang dilengkapi dengan *feedback rotary encoder* untuk mengurangi *error* yang terjadi. Pemilihan arduino menjadi *controller* disebabkan arduino adalah sebuah *single board microcontroller* yang bersifat *open*

source dan populer saat ini. Populer karena memiliki banyak keunggulan dibandingkan beberapa produk *microcontroller* lainnya, seperti mudahnya untuk diimplementasikan dengan perangkat lain baik dalam hal *hardware* atau *software* yang *open source* dan harga yang relatif murah.

2.2 *Microcontroller*

Microcontroller [7] adalah sebuah sistem komputer fungsional dalam sebuah *chip*. Di dalamnya terkandung sebuah inti prosesor, memori (sejumlah kecil RAM, memori program, atau keduanya), dan perlengkapan input output. Dengan kata lain, *microcontroller* adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus, cara kerja *microcontroller* sebenarnya membaca dan menulis data. *Microcontroller* merupakan komputer didalam *chip* yang digunakan untuk mengontrol peralatan elektronik, yang menekankan efisiensi dan efektifitas biaya. Secara harfiahnya bisa disebut “pengendali kecil” dimana sebuah sistem elektronik yang sebelumnya banyak memerlukan komponen-komponen pendukung seperti IC TTL dan CMOS dapat direduksi/diperkecil dan akhirnya terpusat serta dikendalikan oleh *microcontroller* ini.

Ada beberapa jenis *microcontroller* Arduino berdasarkan fungsinya, seperti Arduino UNO, Arduino SERIAL, Arduino MEGA, Arduino FIO, Arduino LILYPAD, Arduino BT, dan Arduino NANO. Dan *microcontroller* pada perancangan kali ini menggunakan Arduino MEGA.

Arduino MEGA adalah papan *microcontroller* berbasis ATmega2560. Arduino MEGA memiliki 54 pin digital input/output, dimana 15 pin dapat digunakan sebagai output PWM, 16 pin sebagai input analog, dan 4 pin sebagai UART (*port serial hardware*), 16 MHz kristal osilator, koneksi USB, *jack power*, header ICSP, dan tombol reset.

Gambar 2.1 dibawah merupakan ilustrasi dari board arduino MEGA, dan tabel 2.1 menjelaskan tentang deskripsi arduino MEGA.



Gambar 2. 1 *Board* Arduino UNO
(www.arduino.cc)

Tabel 2. 1 Deskripsi Arduino MEGA

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
Length	101.52 mm
Width	53.3 mm
Weight	37 g

(www.arduino.cc)

2.2.1 Daya (*Power*)

Board Arduino Mega 2560 dapat ditenagai dengan power yang diperoleh dari koneksi kabel USB, atau via power supply eksternal. Pilihan power yang digunakan akan dilakukan secara otomatis. External power supply dapat diperoleh dari adaptor AC-DC atau bahkan baterai, melalui jack DC yang tersedia, atau menghubungkan langsung GND dan pin Vin yang ada di board. Board dapat beroperasi dengan power dari external power supply yang memiliki tegangan antara 6V hingga 20V. Namun ada beberapa hal yang harus anda perhatikan dalam rentang tegangan ini. Jika diberi tegangan kurang dari 7V, pin 5V tidak akan memberikan nilai murni 5V, yang mungkin akan membuat rangkaian bekerja dengan tidak sempurna. Jika diberi tegangan lebih dari 12V, regulator tegangan bisa over heat yang pada akhirnya bisa merusak pcb. Dengan demikian, tegangan yang di rekomendasikan adalah 7V hingga 12V

Beberapa pin power pada Arduino Uno :

- GND. Ini adalah ground atau negatif.
- Vin. Ini adalah pin yang digunakan jika anda ingin memberikan power langsung ke board Arduino dengan rentang tegangan yang disarankan 7V - 12V.
- Pin 5V. Ini adalah pin output dimana pada pin tersebut mengalir tegangan 5V yang telah melalui regulator.
- 3V3. Ini adalah pin output dimana pada pin tersebut disediakan tegangan 3.3V yang telah melalui regulator
- IOREF. Ini adalah pin yang menyediakan referensi tegangan mikrokontroller. Biasanya digunakan pada board shield untuk memperoleh tegangan yang sesuai, apakah 5V atau 3.3V

2.2.2 Input dan Output

Masing-masing dari 54 digital pin pada Arduino MEGA dapat digunakan sebagai input atau output, menggunakan fungsi *pinMode()* , *digitalWrite()* , dan *digitalRead()*. Arduino MEGA beroperasi pada tegangan 5 volt. Setiap pin dapat memberikan atau menerima arus maksimum 40 mA dan memiliki resistor *pull-up internal* (yang terputus secara default) sebesar 20-50 kOhms. Selain itu, beberapa pin memiliki fungsi khusus, antara lain:

- *Serial* : 0 (RX) dan 1 (TX); *Serial 1* : 19 (RX) dan 18 (TX); *Serial 2* : 17 (RX) dan 16 (TX); *Serial 3* : 15 (RX) dan 14 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data *serial* TTL. Pins 0 dan 1 juga terhubung ke pin chip ATmega16U2 *Serial* USB-to-TTL.
- *Eksternal Interupsi* : Pin 2 (*interrupt* 0), pin 3 (*interrupt* 1), pin 18 (*interrupt* 5), pin 19 (*interrupt* 4), pin 20 (*interrupt* 3), dan pin 21 (*interrupt* 2). Pin ini dapat dikonfigurasi untuk memicu sebuah interupsi pada nilai yang rendah, meningkat atau menurun, atau perubah nilai.
- *SPI* : Pin 50 (MISO), pin 51 (MOSI), pin 52 (SCK), pin 53 (SS). Pin ini mendukung komunikasi SPI menggunakan *library* SPI. Pin SPI juga terhubung dengan header ICSP, yang secara fisik kompatibel dengan Arduino UNO, Arduino Duemilanove dan Arduino Diecimila.
- *LED* : Pin 13. Tersedia secara built-in pada papan Arduino ATmega2560. LED terhubung ke pin digital 13. Ketika pin diset bernilai *HIGH*, maka LED menyala (*ON*), dan ketika pin diset bernilai *LOW*, maka LED padam (*OFF*).
- *TWI* : Pin 20 (SDA) dan pin 21 (SCL). Yang mendukung komunikasi TWI menggunakan *library*

Wire. Perhatikan bahwa pin ini tidak di lokasi yang sama dengan pin TWI pada Arduino Duemilanove atau Arduino Diecimila.

Arduino Mega2560 memiliki 16 pin sebagai analog input, yang masing-masing menyediakan resolusi 10 bit (yaitu 1024 nilai yang berbeda). Secara default pin ini dapat diukur/diatur dari mulai Ground sampai dengan 5 Volt, juga memungkinkan untuk mengubah titik jangkauan tertinggi atau terendah mereka menggunakan pin AREF dan fungsi *analogReference()*.

- AREF. Referensi tegangan untuk input analog. Digunakan dengan *analogReference()*.
- Reset. Membawa saluran ini *LOW* untuk mereset *microcontroller*. Secara khusus, digunakan untuk menambahkan sebuah tombol reset untuk melindungi yang mem-*block* sesuatu pada *board*.

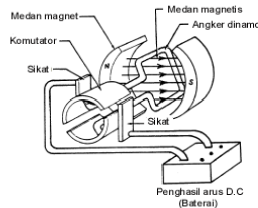
2.3 Motor DC

2.3.1 Pengertian Motor DC

Motor listrik [8] merupakan perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik. Energi mekanik ini digunakan untuk, misalnya memutar *impeller* pompa, *fan* atau *blower*, menggerakkan kompresor, mengangkat bahan, dll. Motor listrik digunakan juga di rumah (*mixer*, bor listrik, kipas angin) dan di industri. Motor listrik kadangkala disebut “kuda kerja” nya industri sebab diperkirakan bahwa motor-motor menggunakan sekitar 70% beban listrik total di industri.

Motor DC memerlukan *supply* tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Kumparan medan pada *motor DC* disebut *stator* (bagian yang tidak berputar) dan kumparan jangkar disebut *rotor* (bagian yang berputar). Jika terjadi putaran pada kumparan jangkar dalam pada medan magnet, maka akan

timbul tegangan (GGL) yang berubah-ubah arah pada setiap setengah putaran, sehingga merupakan tegangan bolak-balik. Prinsip kerja dari arus searah adalah membalik fasa tegangan dari gelombang yang mempunyai nilai positif dengan menggunakan komutator, dengan demikian arus yang berbalik arah dengan *rotor* yang berputar dalam medan magnet. Bentuk motor paling sederhana memiliki kumparan satu lilitan yang bisa berputar bebas di antara kutub-kutub magnet permanen yang ditampilkan pada gambar 2.2 dibawah ini.

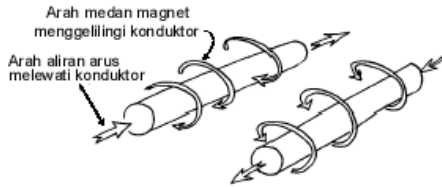


Gambar 2. 2 Motor *DC* Sederhana
(www.academia.edu)

Catu tegangan DC dari baterai menuju ke lilitan melalui sikat yang menyentuh *komutator*, dua segmen yang terhubung dengan dua ujung lilitan. Kumparan satu lilitan pada gambar di atas disebut angker dinamo. Angker dinamo adalah sebutan untuk komponen yang berputar di antara medan magnet.

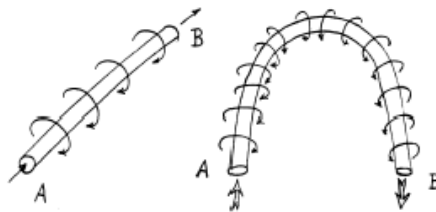
2.3.2 Prinsip Dasar Cara Kerja

Jika arus lewat pada suatu konduktor, timbul medan magnet di sekitar konduktor. Arah medan magnet ditentukan oleh arah aliran arus pada konduktor. Gambar 2.3 berikut ini menggambarkan medan magnet yang membawa arus listrik mengelilingi konduktor.



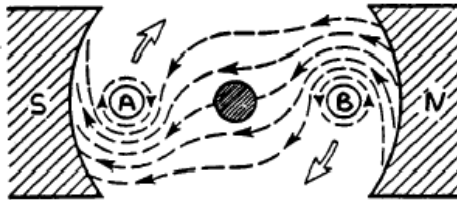
Gambar 2. 3 Medan Magnet yang Membawa Arus Mengelilingi Konduktor
(www.academia.edu)

Aturan genggam tangan kanan dapat dipakai untuk menentukan arah garis fluks di sekitar konduktor. Genggam konduktor dengan tangan kanan dengan jempol mengarah pada arah aliran arus, maka jari-jari anda akan menunjukkan arah garis fluks. Gambar 2.4 menunjukkan medan magnet yang terbentuk di sekitar konduktor berubah arah karena bentuk U.



Gambar 2. 4 Medan Magnet yang Membawa Arus Mengelilingi Konduktor U
(www.academia.edu)

Jika konduktor berbentuk U (angker dinamo) diletakkan di antara kutub utara dan selatan yang kuat medan magnet konduktor akan berinteraksi dengan medan magnet kutub.



Gambar 2. 5 Reaksi Garis Fluks
(www.academia.edu)

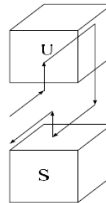
Gambar 2.5 diatas merupakan reaksi garis fluks, lingkaran bertanda A dan B merupakan ujung konduktor yang dilengkungkan (*looped conductor*). Arus mengalir masuk melalui ujung A dan keluar melalui ujung B. Medan konduktor A yang searah jarum jam akan menambah medan pada kutub dan menimbulkan medan yang kuat di bawah konduktor. Konduktor akan berusaha bergerak ke atas untuk keluar dari medan kuat ini. Medan konduktor B yang berlawanan arah jarum jam akan menambah medan pada kutub dan menimbulkan medan yang kuat di atas konduktor. Konduktor akan berusaha untuk bergerak turun agar keluar dari medan yang kuat tersebut. Gaya-gaya tersebut akan membuat angker dinamo berputar searah jarum jam.

Mekanisme kerja untuk seluruh jenis motor secara umum :

- Arus listrik dalam medan magnet akan memberikan gaya.
- Jika kawat yang membawa arus dibengkokkan menjadi sebuah lingkaran / loop, maka kedua sisi loop, yaitu pada sudut kanan medan magnet, akan mendapatkan gaya pada arah yang berlawanan.
- Pasangan gaya menghasilkan tenaga putar / *torque* untuk memutar kumparan

- Motor-motor memiliki beberapa loop pada dinamonya untuk memberikan tenaga putaran yang lebih seragam dan medan magnetnya dihasilkan oleh susunan elektromagnetik yang disebut kumparan medan

Pada motor DC, daerah kumparan medan yang dialiri arus listrik akan menghasilkan medan magnet yang melingkupi kumparan jangkar dengan arah tertentu. Konversi dari energi listrik menjadi energi mekanik (motor) maupun sebaliknya berlangsung melalui medan magnet, dengan demikian medan magnet disini selain berfungsi sebagai tempat untuk menyimpan energi, sekaligus sebagai tempat berlangsungnya proses perubahan energi, daerah tersebut dapat dilihat pada gambar 2.6 di bawah ini :



Gambar 2. 6 Prinsip Kerja Motor DC
(www.academia.edu)

Agar proses perubahan energi mekanik dapat berlangsung secara sempurna, maka tegangan sumber harus lebih besar daripada tegangan gerak yang disebabkan reaksi lawan. Dengan memberi arus pada kumparan jangkar yang dilindungi oleh medan maka menimbulkan perputaran pada motor.

Dalam memahami sebuah motor, penting untuk mengerti apa yang dimaksud dengan beban motor. Beban

dalam hal ini mengacu kepada keluaran tenaga putar / *torque* sesuai dengan kecepatan yang diperlukan. Beban umumnya dapat dikategorikan ke dalam tiga kelompok :

- Beban *torque* konstan adalah beban dimana permintaan keluaran energinya bervariasi dengan kecepatan operasinya namun *torque*-nya tidak bervariasi. Contoh beban dengan *torque* konstan adalah *conveyors*, *rotary kilns*, dan pompa *displacement* konstan.
- Beban dengan variabel *torque* adalah beban dengan *torque* yang bervariasi dengan kecepatan operasi. Contoh beban dengan variabel *torque* adalah pompa *sentrifugal* dan *fan* (*torque* bervariasi sebagai kuadrat kecepatan)
- Beban dengan energi konstan adalah beban dengan permintaan *torque* yang berubah dan berbanding terbalik dengan kecepatan. Contoh untuk beban dengan daya konstan adalah peralatan-peralatan mesin.

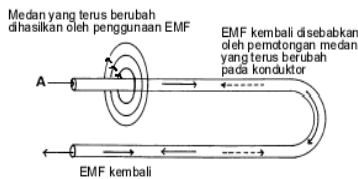
2.3.3 Prinsip Arah Putaran Motor

Untuk menentukan arah putaran motor digunakan kaedah *Flamming* tangan kiri. Kutub-kutub magnet akan menghasilkan medan magnet dengan arah dari kutub utara ke kutub selatan. Jika medan magnet memotong sebuah kawat penghantar yang dialiri arus searah dengan empat jari, maka akan timbul gerak searah ibu jari. Gaya ini disebut gaya *Lorentz*, yang besarnya sama dengan F .

Prinsip motor: aliran arus di dalam penghantar yang berada di dalam pengaruh medan magnet akan menghasilkan gerakan. Besarnya gaya pada penghantar akan bertambah besar jika arus yang melalui penghantar bertambah besar.

2.3.4 *Electromotive Force (EMF) / Gaya Gerak Listrik*

EMF induksi biasanya disebut *EMF Counter*. *EMF counter* artinya adalah EMF tersebut ditimbulkan oleh angker dinamo yang melawan tegangan yang diberikan padanya. Teori dasarnya adalah jika sebuah konduktor listrik memotong garis medan magnet maka timbul GGL pada konduktor, digambarkan pada gambar 2.7 dibawah ini.



Gambar 2. 7 *EMF Counter*
(www.academia.edu)

EMF induksi terjadi pada motor listrik, generator serta rangkaian listrik dengan arah berlawanan terhadap gaya yang menimbulkannya. HF. Emil Lenz mencatat pada tahun 1834 bahwa “arus induksi selalu berlawanan arah dengan gerakan atau perubahan yang menyebabkannya”. Hal ini disebut sebagai Hukum Lenz.

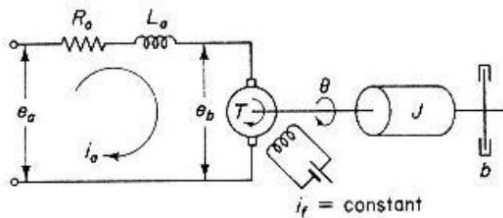
Timbulnya EMF tergantung pada:

- kekuatan garis *fluks* magnet
- jumlah lilitan konduktor
- sudut perpotongan *fluks* magnet dengan konduktor
- kecepatan konduktor memotong garis *fluks* magnet

Tidak ada arus induksi yang terjadi jika angker dinamo diam.

2.3.5 Model Matematis Motor DC

Sistem elektromekanika merupakan sistem gabungan antara elektrik dengan mekanik. Sistem elektromekanika yang akan dibahas adalah motor DC. Motor DC dengan daya rendah, biasanya digunakan untuk komputer dan instrumentasi seperti misalnya *disk drive*, *tape drive*, dan lain lain. Sedangkan yang berdaya menengah digunakan untuk sistem lengan robot, mesin pemintal yang dikendalikan secara numerik. Diagram skematik motor DC ditunjukkan pada gambar 2.8 dibawah ini.



Gambar 2. 8 Diagram Skematik Motor DC
(www.academia.edu)

Bila kumparan magnet berputar, maka tegangan akan sebanding dengan hasil kali *fluks* dan kecepatan sudut yang diinduksikan pada kumparan magnet. Untuk *fluks* yang konstan, tegangan induksi e_b berbanding lurus dengan kecepatan sudut $d\theta/dt$. Persamaan tersebut ditunjukkan oleh persamaan (2.1) dibawah ini.

$$e_b = K_b \frac{d\theta}{dt} \quad (2.1)$$

Kecepatan jangkar magnet motor DC dikendalikan oleh tegangan kumparan magnet. Persamaan diferensial rangkaian kumparan magnet ditunjukkan oleh persamaan (2.2) berikut

$$L_a \frac{di_a}{dt} + R_a i_a + eb = ea \quad (2.2)$$

Arus jangkar magnet menghasilkan torsi yang bekerja terhadap inersia dan gesekan. Persamaan tersebut ditunjukkan oleh persamaan (2.3) dibawah ini.

$$J \frac{d^2 \theta}{dt^2} + b \frac{d\theta}{dt} = T = K_t i_a \quad (2.3)$$

Dengan transformasi Laplace ketiga persamaan terakhir (2.1), (2.2) dan (2.3) dapat dirubah sebagai berikut,

$$E_b(s) = K_b s \theta(s) \quad (2.4)$$

$$(L_a s + R_a) I_a(s) + E_b(s) = E_a(s) \quad (2.5)$$

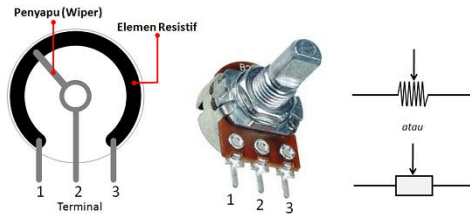
$$(J s^2 + B s) \theta(s) = T(s) = K_t I_a(s) \quad (2.6)$$

Dengan menyusun persamaan (2.4), (2.5) dan (2.6) diperoleh *transfer function*,

$$\frac{\theta(s)}{E_a(s)} = \frac{K_t}{L_a J s^3 + (L_a B + R_a J) s^2 + (R_a B + K_t K_b) s} \quad (2.7)$$

2.4 Potensiometer

Potensiometer adalah salah satu jenis Resistor yang nilai resistansinya dapat diatur sesuai dengan kebutuhan rangkaian elektronika ataupun kebutuhan pemakainya. Potensiometer merupakan keluarga resistor yang tergolong dalam kategori variable resistor. Secara struktur, potensiometer terdiri dari 3 kaki terminal dengan sebuah shaft atau tuas yang berfungsi sebagai pengaturnya. Gambar 2.9 dibawah ini menunjukkan struktur internal potensiometer beserta bentuk dan simbolnya.



Gambar 2. 9 Struktur Potensiometer beserta Bentuk dan Simbolnya
(www.elektronika.com)

Pada dasarnya bagian-bagian penting dalam Komponen Potensiometer adalah :

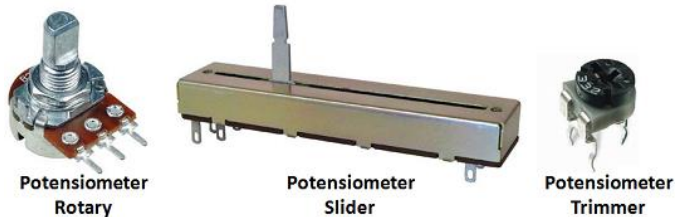
1. Penyapu atau disebut juga dengan Wiper
2. Element Resistif
3. Terminal

2.4.1 Jenis-Jenis Potensiometer

Berdasarkan bentuknya, Potensiometer dapat dibagi menjadi 3 macam, yaitu :

1. Potensiometer Slider, yaitu potensiometer yang nilai resistansinya dapat diatur dengan cara menggeserkan wiper dari kiri ke kanan atau dari bawah ke atas sesuai dengan pemasangannya.
2. Potensiometer Rotary, yaitu potensiometer yang nilai resistansinya dapat diatur dengan cara memutar wiper sepanjang lintasan yang melingkar. Potensiometer Rotary sering disebut juga dengan *Thumbwheel Potentiometer*.
3. Potensiometer Trimmer, yaitu potensiometer yang bentuknya kecil dan harus menggunakan alat khusus seperti *screwdriver* untuk memutarnya. Potensiometer Trimmer ini biasanya dipasangkan di PCB dan jarang dilakukan pengaturannya.

Ketiga jenis potensiometer diatas ditunjukkan pada gambar 2.10 dibawah ini.



Gambar 2. 10 Jenis-Jenis Potensiometer
(www.elektronika.com)

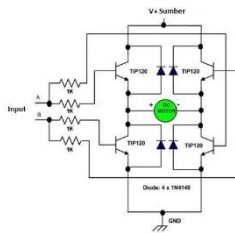
2.4.2 Prinsip Kerja Potensiometer

Sebuah potensiometer terdiri dari sebuah elemen resistif yang membentuk jalur atau *track* dengan terminal di kedua ujungnya. Sedangkan terminal lainnya (biasanya berada di tengah) adalah penyapu atau *wiper* yang dipergunakan untuk menentukan pergerakan pada jalur elemen resistif atau *resistive*. Pergerakan penyapu (*wiper*) pada jalur elemen resistif inilah yang mengatur naik-turunnya nilai resistansi sebuah potensiometer. Elemen resistif pada potensiometer umumnya terbuat dari bahan campuran logam dan keramik ataupun bahan karbon.

2.5 Driver Motor DC H-Bridge

Driver motor DC tipe *H-Bridge* [9] menggunakan power driver berupa transistor. Rangkaian *driver motor DC H-Bridge* transistor ini dapat mengendalikan arah putaran *motor DC* dalam 2 arah dan dapat dikontrol dengan metode PWM (*Pulse Width Modulation*) maupun metode sinyal logika dasar TTL (*high*) dan (*low*). Untuk pengendalian *motor DC* dengan metode PWM maka dengan rangkaian *driver motor DC* ini kecepatan putaran *motor DC* dapat dikendalikan dengan baik. Apabila menggunakan metode logika TTL 0 dan 1 maka

rangkaian ini hanya dapat mengendalikan arah putaran *motor DC* saja dengan kecepatan putaran *motor DC* maksimum. Rangkaian *driver motor DC H-Bridge* ini menggunakan rangkaian jembatan transistor 4 unit dengan proteksi impuls tegangan induksi *motor DC* berupa dioda yang dipasang paralel dengan masing-masing transistor secara *reverse bias*. Rangkaian *driver motor DC* secara detail dapat dilihat pada gambar 2.11 berikut.



Gambar 2. 11 Rangkaian *Driver Motor DC H-Bridge* Transistor
(<http://elektronika-dasar.web.id>)

Proses mengendalikan *motor DC* menggunakan rangkaian *driver motor DC H-Bridge* diatas dapat diuraikan dalam beberapa bagian sebagai berikut: *Driver Motor DC* dengan metode logika TTL (0 dan 1) atau *high* dan *low* hanya dapat mengendalikan arah putar *motor DC* dalam 2 arah tanpa pengendalian kecepatan putaran (kepatan maksimum). untuk mengendalikan *motor DC* dalam 2 arah dengan rangkaian *driver motor DC H-bridge* diatas konfigurasi kontrol pada jalur input adalah dengan memberikan input berupa logika TTL ke jalur input A dan B.

- Untuk mengendalikan arah putar searah jarum jam adalah dengan memberikan logika TTL 1 (*high*) pada jalur input A dan logika TTL 0 (*low*) pada jalur input B.

- Untuk mengendalikan arah putar berlawanan arah jarum jam adalah dengan memberikan logika TTL 1 (*high*) pada jalur input B dan logika TTL 0 (*low*) pada jalur input A. Driver *motor DC* dengan metode PWM (*Pulse Width Modulation*) dapat mengendalikan arah putaran *motor DC* dan kecepatan *motor DC* menggunakan pulsa PWM yang diberikan ke jalur input A dan B, dimana konfigurasi sinyal kontrol sebagai berikut.
- Untuk mengendalikan arah putar *motor DC* searah jarum jam dengan kecepatan dikendalikan pulsa PWM maka jalur input B selalu diberikan logika TTL 0 (*low*) dan jalur input A diberikan pulsa PWM.
- Untuk mengendalikan arah putar *motor DC* berlawanan arah jarum jam dengan kecepatan dikendalikan pulsa PWM maka jalur input A selalu diberikan logika TTL 0 (*low*) dan jalur input B diberikan pulsa PWM.

Kecepatan putaran *motor DC* dikendalikan oleh persentasi *Ton-duty cycle* pulsa PWM yang diberikan ke jalur input rangkaian *driver motor DC H-bridge* transistor diatas

2.6 Analisa Kestabilan

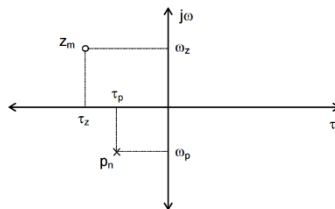
Hal yang amat penting dalam desain sistem kontrol adalah masalah stabilitas system [2]. Bukan hal yang rahasia lagi bahwa pokok tujuan terpenting dalam analisa dan desain kontrol adalah menciptakan suatu sistem yang stabil. Suatu sistem dikatakan stabil apabila tercipta kondisi dimana tanggapan (*response*) sistem bersifat terbatas jika diberikan masukan sistem yang bersifat terbatas pula. yang dimaksud terbatas di sini adalah harga maksimumnya terbatas, sebagai contoh: $f(t) = A \sin \omega t$, maka harga maksimum dari $f(t)$ tidak akan melebihi A (terbatas pada A)

2.6.1 Konsep Umum Kestabilan

Bila diberikan persamaan fungsi alih dari suatu sistem loop tertutup sebagai berikut:

$$\frac{C(s)}{R(s)} = \frac{(s+z_1)(s+z_2)\dots(s+z_m)}{(s+p_1)(s+p_2)\dots(s+p_n)}, \text{ dimana } n > m \quad (2.8)$$

maka akar-akar dari persamaan pembilang (*nominator*) yaitu $s = -z_1, s = -z_2, \dots, s = -z_n$ adalah *zero* dari fungsi alih *loop* tertutup sistem. Sedangkan akar-akar dari persamaan penyebut (*denominator*) yaitu $s = -p_1, s = -p_2, \dots, s = -p_m$ adalah *zero-zero* dari fungsi alih *loop* tertutup sistem. Dalam penggambaran pada bidang s (koordinat *real* vs *imajiner*), *zero* digambarkan dengan tanda bulatan kecil dan *pole* digambarkan dengan tanda silang. Hal ini dapat dilihat pada gambar 2.12 dibawah ini.



Gambar 2. 12 Letak *Pole* dan *Zero* dalam Bidang s
(www.scribd.com)

Dari Gambar 2.21 *zero* z_m dinyatakan dalam koordinat $z_m (\tau_z, \omega_z)$ dimana nilai dari $z_m = \tau_z + j\omega_z$. *Pole* p_n dinyatakan dalam koordinat $p_n (\tau_p, \omega_p)$ dimana nilai dari $p_n = \tau_p + j\omega_p$.

Dalam analisa kestabilan, seringkali yang digunakan adalah akar-akar dari persamaan *denominator* yaitu *pole-pole*. Karena seringkali penggunaan persamaan

denominator ini untuk menganalisa karakteristik kestabilan suatu sistem, maka persamaan *denominator* ini diberi nama persamaan karakteristik.

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1+G(s)H(s)} \quad (2.9)$$

$$1 + G(s)H(s) = 0 \quad (2.10)$$

2.6.1.1 Analisa Kestabilan dari Letak *Pole-Pole*

Fungsi alih loop tertutup dapat diubah menjadi deret parsial sebagai berikut :

$$\frac{C(s)}{R(s)} = \frac{A_0}{s-p_0} + \frac{A_1}{s-p_1} + \dots + \frac{A_n}{s-p_n} \quad (2.11)$$

Dimana:

$A_0 \dots A_n$ = konstanta

$p_0 \dots p_n$ = akar-akar persamaan karakteristik

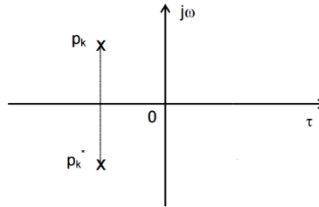
Tanggapan waktu untuk input unit *step* adalah :

$$c(t) = 1 + A_0 e^{p_0 t} + A_1 e^{p_1 t} + \dots + A_n e^{p_n t} \quad (2.12)$$

Bentuk umum akar-akar persamaan karakteristik diberikan oleh :

$$p_k = \tau_k + j\omega_k \quad (2.13)$$

Kestabilan suatu sistem dapat dilihat dari letak *pole-pole* fungsi alih *loop* tertutup atau akar-akar dari persamaan karakteristik pada bidang s . Bila semua τ_k negatif, letak salah satu akar p_k pada bidang s terlihat seperti gambar 2.13, dimana p_k^* adalah konjugate dari p_k .



Gambar 2. 13 Pole dengan τ_k Negatif
(www.scribd.com)

Tanggapan waktu dari masukan unit step adalah :

$$|c(t)| = 1 + |A_0||e^{\tau_0 t}| + \dots + |A_n||e^{\tau_n t}| \quad (2.14)$$

untuk setiap t , maka :

$$|e^{j\omega_n t}| = 1 \text{ dan } |e^{\tau_k t}| < \infty \quad (2.15)$$

$|e^{\tau_k t}|$ untuk τ_k negatif akan bernilai satu bila $t = 0$ dan akan bergerak turun hingga nol untuk waktu tak terhingga. Dengan demikian nilai dari $|c(t)|$ selalu kurang dari tak terhingga, dengan artian nilainya terbatas.

Konsep stabilitas adalah adanya keterbatasan nilai tanggapan terhadap masukan yang bersifat terbatas. Dalam kondisi ini (τ_k negatif), syarat stabilitas terpenuhi, sehingga dapat disimpulkan: bila suatu sistem mempunyai akar-akar persamaan karakteristik (*pole-pole*) yang terletak di sebelah kiri sumbu *imaginer* pada bidang s , maka sistem tersebut adalah stabil.

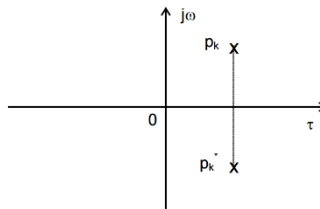
Bila salah satu dari τ_k ada yang bernilai positif, letak salah satu akar p_k pada bidang s terlihat seperti gambar 2.14, dimana p_k^* adalah *konjugate* dari p_k . Tanggapan waktu dari masukan unit step adalah :

$$|c(t)| = 1 + |A_0||e^{\tau_0 t}| + \dots + |A_n||e^{\tau_n t}| \quad (2.16)$$

untuk setiap t , maka untuk nilai τ_k negatif :

$$|e^{\tau_n t}| < \infty \quad (2.17)$$

sedangkan untuk nilai τ_k positif, nilainya akan satu untuk $t = 0$ dan tak berhingga untuk waktu yang tak terhingga. Sehingga nilai $|c(t)|$ menjadi tak berhingga untuk waktu tak terhingga, dengan demikian tanggapan waktunya menjadi tak terbatas.



Gambar 2. 14 Pole dengan τ_k Positif
(www.scribd.com)

Karena tanggapan waktu masukan fungsi *step* yang tak terbatas, maka dapat disimpulkan: bila ada salah satu atau lebih akar-akar dari persamaan karakteristik (*pole-pole*) terletak di sebelah kanan sumbu *imaginer* pada bidang s , maka sistem tersebut tidak stabil.

Analisa kestabilan dengan menggunakan persamaan karakteristik ini cukup mudah dan tidak berbelit-belit. Akan tetapi cara ini menjadi semakin sulit untuk diaplikasikan pada sistem-sistem berorde tinggi, karena untuk mencari akar-akar persamaan *polinomial* derajat n (untuk $n > 2$) tidak semudah mencari akar-akar persamaan kuadrat biasa. Dengan sulitnya mencari akar-akar persamaan tersebut, penggunaan persamaan karakteristik ini menjadi tidak efisien lagi. Untuk itu beberapa analisa kestabilan yang lain akan dibahas dalam bab ini.

2.6.1.2 Kriteria Stabilitas Hurwitz

Persamaan karakteristik sistem orde n diberikan oleh:

$$a_0 s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n = 0 \quad (2.18)$$

Dari persamaan karakteristik tersebut dapat dibentuk suatu matrik determinan yang sering disebut sebagai determinan Hurwitz sebagai berikut :

$$\begin{vmatrix} a_1 & a_0 & 0 & \cdots & 0 \\ a_3 & a_2 & a_1 & \cdots & 0 \\ a_5 & a_4 & a_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{2n-1} & a_{2n-2} & a_{2n-3} & \cdots & a_n \end{vmatrix} \rightarrow \text{determinan Hurwitz}$$

Nilai-nilai untuk koefisien dengan indeks lebih besar dari n atau dengan indeks negatif diganti dengan nol.

Kondisi stabilitas terpenuhi jika :

$$\Delta_k > 0, \text{ untuk } k = 1, 2, 3, \dots, n \quad (2.19)$$

dimana nilai-nilai Δ_k dihitung dengan cara :

$$\Delta_1 = a_1 \quad (2.20)$$

$$\Delta_2 = \begin{vmatrix} a_1 & a_0 \\ a_3 & a_2 \end{vmatrix} \quad (2.21)$$

$$\Delta_3 = \begin{vmatrix} a_1 & a_0 & 0 \\ a_3 & a_2 & a_1 \\ a_5 & a_4 & a_3 \end{vmatrix} \quad (2.22)$$

hingga Δ_n .

Jadi, syarat sistem stabil bila keseluruhan nilai dari determinan-determinan tersebut adalah positif.

2.6.1.3 Kriteria Stabilitas Routh-Hurwitz

Persamaan karakteristik sistem orde n diberikan oleh:

$$a_0 s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n = 0 \quad (2.23)$$

Dari persamaan karakteristik tersebut dapat dibentuk suatu matrik atau deret (*Routh array*) dimana hanya dua baris teratas saja yang ditentukan langsung dari persamaan karakteristiknya.

$$\begin{array}{c|cccccc}
 s^n & a_0 & a_2 & a_4 & a_6 & \cdots & 0 \\
 s^{n-1} & a_1 & a_3 & a_5 & a_7 & \cdots & 0 \\
 s^{n-2} & b_1 & b_2 & b_3 & & & \\
 s^{n-3} & c_1 & c_2 & c_3 & & & \\
 \vdots & \vdots & & & & & \\
 s^1 & \vdots & & & & & \\
 s^0 & h_1 & & & & &
 \end{array} \rightarrow \text{deret Routh}$$

dimana :

$$b_1 = \frac{a_1 a_2 - a_0 a_3}{a_1} \quad (2.24)$$

$$b_2 = \frac{a_1 a_4 - a_0 a_5}{a_1} \quad (2.25)$$

$$b_3 = \frac{a_1 a_6 - a_0 a_7}{a_1} \quad (2.26)$$

sampai nol.

Kemudian :

$$c_1 = \frac{b_1 a_3 - a_1 b_2}{b_1} \quad (2.27)$$

$$c_2 = \frac{b_1 a_5 - a_1 b_3}{b_1} \quad (2.28)$$

sampai semua koefisien didapat sehingga membentuk matrik setengah piramida terbalik.

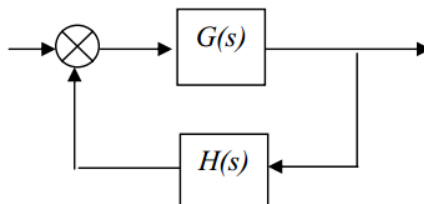
Syarat kestabilan dari analisa *Routh-Hurwitz* ini adalah bila semua koefisien dari kolom pertama deret *Routh* bernilai positif. Bila ada salah satu atau lebih dari koefisien-koefisien tersebut bernilai negatif, maka sistem tersebut tidak stabil. Jumlah akar-akar positif dari persamaan karakteristik sebanding dengan jumlah perubahan tanda (dari positif ke negatif atau sebaliknya) pada kolom pertama tersebut

2.6.1.4 Kriteria Stabilitas Nyquist

Sistem loop tertutup yang dibahas seperti ditunjukkan dalam gambar 2.15 dibawah ini. Persamaan karakteristik sistem diberikan oleh persamaan 2.29 dibawah ini

$$1 + G(s)H(s) = 0 \quad (2.29)$$

Persamaan ini menentukan stabilitas sistem, dan jika sistem dalam keadaan stabil, maka karakteristik tanggapan transien sangat bermanfaat. Diagram Bode fungsi alih *loop* terbuka dapat diplot menggunakan fungsi alih $G(j\omega)H(j\omega)$. Penentuan stabilitas sistem *loop* tertutup dari diagram Bode fungsi *loop* terbuka $G(j\omega)H(j\omega)$. Metode yang digunakan berdasarkan kriteria *Nyquist*.

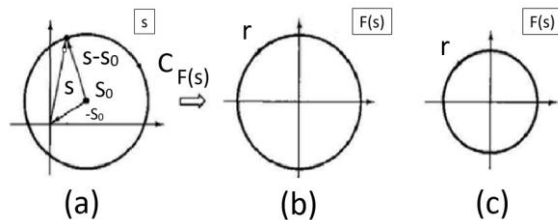


Gambar 2. 15 Sistem *Loop* Tertutup
(www.scribd.com)

Untuk memperkenalkan kriteria *Nyquist*, perlu pemahaman pemetaan (fungsi) dari bidang kompleks s ke bidang $F(s)$. Sebagai contoh kasus fungsi (pemetaan) $F(s)$ diberikan oleh

$$F(s) = s - s_0 \quad (2.30)$$

Dengan s_0 adalah nilai tertentu yang dimungkinkan berupa nilai kompleks. Jika menginginkan memetakan lingkaran berpusat di s_0 dalam bidang s ke dalam bidang $F(s)$, seperti ditunjukkan dalam gambar 2.16 dibawah ini



Gambar 2. 16 Pemetaan Bidang Kompleks
(www.scribd.com)

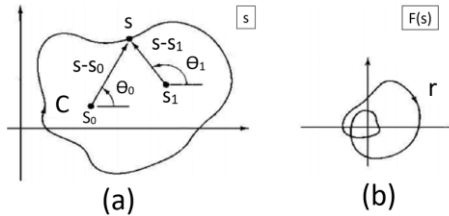
Kurva C dalam bidang s dalam gambar 2.16(a) dipetakan ke kurva Γ dalam bidang $F(s)$ dengan menguji $F(s)$ untuk titik-titik pada kurva C dan melukiskannya nilai-nilai kompleks dalam bidang $F(s)$. Untuk fungsi yang sederhana seperti persamaan (2.30), $F(s)$ merupakan vektor sebagaimana ditunjukkan dalam gambar 2.16(a), dan kurva hasil dalam $F(s)$ terlihat menjadi lingkaran dengan jejari yang sama dengan C tetapi berpusat di titik $(0,0)$.

$$F(s) = \frac{1}{s - s_0} \quad (2.31)$$

Merupakan fungsi terbalik dari persamaan (2.31), jika kurva C dalam gambar 2.16(a) dipetakan ke bidang $F(s)$ melewati persamaan (2.31), vektor $s - s_0$ tetap seperti gambar 2.16(a). Maka $F(s)$ merupakan kebalikan vektor ini. Magnituda $F(s)$ merupakan kebalikan dari yang ditunjukkan dalam gambar 2.16(b), dan sudut berupa negatif, Jadi dalam kasus ini, kurva Γ dalam bidang $F(s)$ juga lingkaran, seperti ditunjukkan dalam gambar 2.16(c), kecuali arah perjalanan sekerang berlawanan dengan jarum jam. Untuk fungsi ini, pelingkupan searah jarum jam suatu *pole* dalam bidang s mengantarkan pelingkupan berlawanan dengan jarum jam suatu titik $(0,0)$ dalam bidang $F(s)$. Pada pemetaan ketiga, jika pemetaan $F(s)$ diberikan oleh

$$F(s) = (s - s_0)(s - s_1) \quad (2.32)$$

Jika bahwa kurva C dalam bidang s melingkupi kedua zero s_0 dan s_1 , seperti dalam gambar 2.17 dibawah. Dalam kasus ini kurva C bukanlah sebuah lingkaran. Dua vektor yang dimiliki oleh $F(s)$ diperlihatkan dalam gambar 2.17(a). Ketika titik s mengelilingi kurva C , maka sudut vektor $(s - s_0)$ berubah sebesar -360° , juga demikian bagi vektor $(s - s_1)$. Oleh karena itu sudut fungsi $F(s)$ berubah -720° . Pada saat bersamaan magnituda dari dua vektor terbatas dan bukan nol. Sehingga kurva Γ haruslah melingkupi titik $(0,0)$ dua kali, seperti diperlihatkan dalam gambar 2.17(b). Perhatikan bahwa kurva C dalam melingkupi dua zero $F(s)$ searah jarum jam. Dalam bidang $F(s)$, kurva Γ melingkupi titik pusat dua kali.



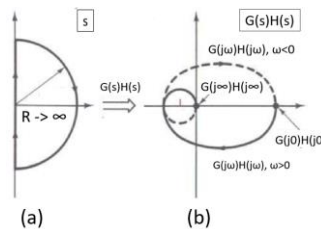
Gambar 2. 17 Jalan Pelingkupan Dua Pole
(www.scribd.com)

Jika pemetaan $F(s)$ merupakan bentuk terbalik dari persamaan (2.32), yaitu jika

$$F(s) = \frac{1}{(s-s_0)(s-s_1)} \quad (2.33)$$

Vektor untuk kurva C masih seperti diperlihatkan gambar 2.17. Karena sudut perkalian dua vektornya berputar -720° , maka sudut $F(s)$ berputar mengelilingi 720° . Oleh karena itu pemetaan ke kurva Γ akan menghasilkan pelingkupan dua kali berlawanan dengan jarum jam.

Kriteria *Nyquist* dapat dinyatakan dengan rujukan pemetaan seperti diperlihatkan dalam gambar 2.18 dibawah ini.



Gambar 2. 18 Diagram *Nyquist*
(www.scribd.com)

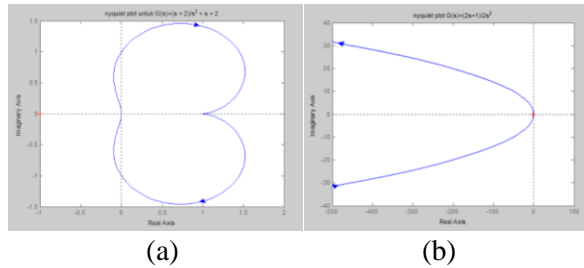
Jalan *Nyquist* ditunjukkan dalam gambar 2.18(a). Jalan ini dipetakan melalui fungsi *loop* terbuka $G(s)H(s)$ ke diagram *Nyquist*, seperti diilustrasikan dalam gambar 2.18(b). Maka

$$Z = N + P \quad (2.34)$$

Dengan Z adalah jumlah akar persamaan karakteristik sistem yang berada setengah kanan bidang kompleks, N adalah jumlah pelingkupan searah jarum jam titik -1 , dan P adalah jumlah *pole* fungsi *loop* terbuka $G(s)H(s)$ yang berada di setengah kanan bidang kompleks. Dalam menguji kestabilan sistem pengendalian dengan menggunakan kriteria kestabilan *nyquist* ada 3 kemungkinan, yaitu:

- Tidak ada pengelilingan titik $-1+j0$. Ini berarti bahwa sistem stabil jika tidak ada kutub dari $G(s)H(s)$ yang terletak disebelah kanan sumbu khayal bidang s , jika tidak demikian sistem tidak stabil.
- Ada satu atau lebih pengelilingan titik $-1+j0$ berlawanan jarum jam. Dalam hal ini sistem stabil jika banyaknya pengelilingan yang berlawanan arah dengan jarum jam sama dengan banyaknya kutub dari $G(s)H(s)$ yang terletak disebelah kanan sumbu khayal bidang s , jika tidak demikian maka sistem tidak stabil.
- Ada satu atau lebih pengelilingan titik $-1+j0$ yang searah jarum jam. Dalam hal ini sistem tidak stabil.

Gambar 2.19 dibawah ini merupakan ilustrasi diagram *nyquist* untuk menunjukkan perbedaan sistem yang stabil dan sistem yang tidak stabil.



Gambar 2. 19 Diagram *Nyquist*
(www.scribd.com)

Gambar 2.19(a) diatas merupakan diagram *nyquist* yang memiliki sistem stabil, karena plot yang terbentuk tidak melingkupi titik $(-1, j0)$, dan gambar 2.19(b) merupakan diagram *nyquist* yang memiliki sistem tidak stabil, karena plot yang terbentuk melingkupi titik $(-1, j0)$.

BAB III

METODOLOGI PERENCANAAN

Metodologi perencanaan bertujuan untuk memberikan gambaran umum proses menganalisa yang dilakukan, dimana terdapat beberapa langkah-langkah perencanaan yang dilakukan adalah sebagai berikut:

1. Studi literatur yang meliputi konsep dasar *microcontroller*, konsep dasar kontrol *flexible fixture*, karakter *flexible fixture*.
2. Tahap pengumpulan data.
3. Desain spesifikasi sistem kontrol.
4. Perencanaan cara kerja *flexible fixture* dengan *microcontroller*.
5. Analisa kontrol *flexible fixture* dengan *microcontroller*.
6. Merancang bangun otomatisasi *flexible fixture* dengan *microcontroller*.

Studi perencanaan *flexible fixture* berbasis *microcontroller* dimana *flexible fixture* yang umumnya digunakan sebagai alat pemegang benda kerja pada mesin-mesin tertentu, seperti mesin milling (*milling machine*), mesin bor (*drilling machine*), mesin miiling CNC (*milling CNC machine*) dan lain-lain, dapat bekerja secara *flexible* dengan mengatur sudut kemiringan dan rotasi sehingga dalam prosesnya akan lebih mudah dan efisien. Perencanaan ini dilakukan untuk mengetahui performa (seperti kontrol gerak *motor DC*) *microcontroller* terhadap *flexible fixture*.

Penggunaan *microcontroller* pada perencanaan ini bertujuan untuk mengontrol pergerakan dari *flexible fixture* sampai mendapatkan pergerakan dari *flexible fixture* yang sesuai dengan yang diinginkan. Hal ini dilakukan dengan cara memprogram *microcontroller* dengan *arduino software (IDE)*.

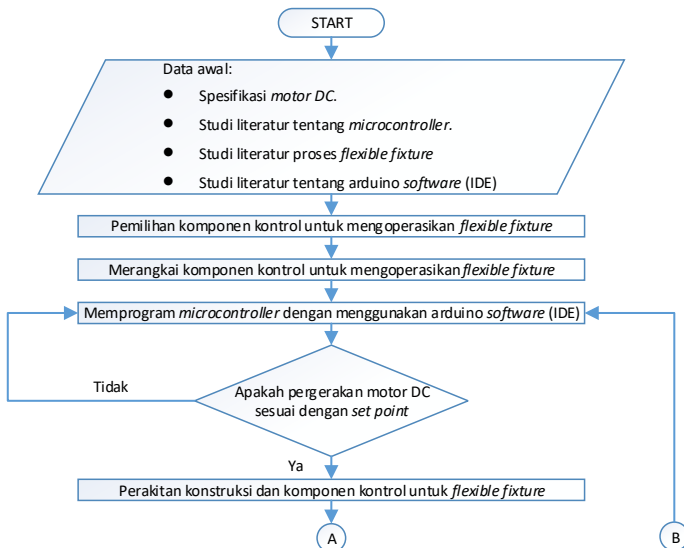
3.1 Tahap Pengumpulan Data

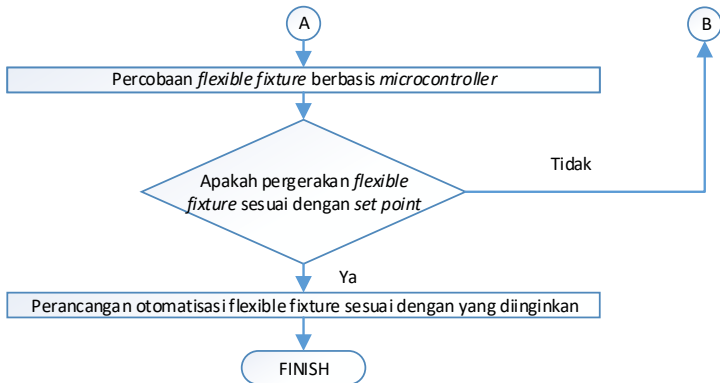
tahap pengumpulan data, diperlukan data-data seperti berikut:

- Spesifikasi *motor DC*.
- Studi literatur tentang *microcontroller*.
- Studi literatur proses *flexible fixture*
- Studi literatur tentang arduino *software* (IDE)

3.2 Tahap Perencanaan otomatisasi *Flexible Fixture*

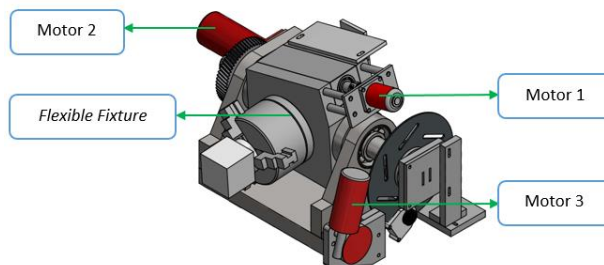
Tahap perancangan *flexible fixture* ini berguna untuk mendapatkan gambaran awal yang dapat digunakan dalam merangkai sistem kontrol *flexible fixture* berbasis *microcontroller*. Dibawah ini adalah gambar 3.1 yang menjelaskan tentang diagram alir dari proses perancangan otomatisasi pada *flexible fixture*.





Gambar 3. 1 Diagram Alir Perencanaan Sistem Kontrol *Flexible Fixture*

Dari diagram alir perencanaan sistem kontrol *flexible fixture* diatas akan dijelaskan tentang langkah-langkah kerja *flexible fixture*. Berikut adalah cara kerja atau langkah-langkah proses kerja dari *flexible fixture* yang ditunjukkan pada gambar 3.2 dibawah ini:

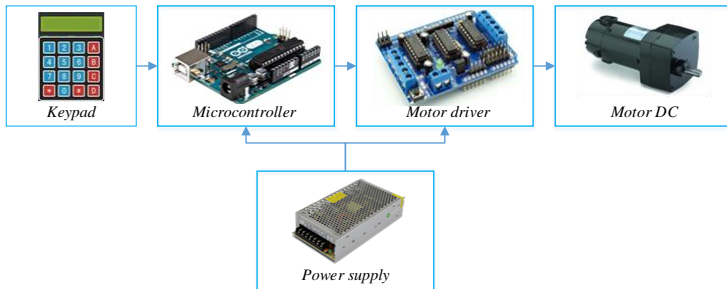


Gambar 3. 2 *Flexible Fixture* yang Direncanakan

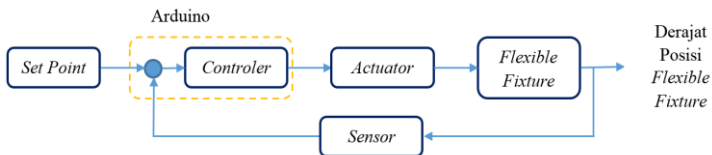
- Menghidupkan saklar *power supply* untuk *microcontroller* dan *motor DC driver*.

- Pada P1, memutar potensiometer untuk memasukkan posisi sudut yang diinginkan motor 1.
- Pada P2, memutar potensiometer untuk memasukkan posisi sudut yang diinginkan motor 2.
- Menekan tombol RUN, setelah posisi sudut motor 2 telah tercapai motor 3 akan berputar untuk mengaktifkan rem pengunci, kemudian motor 1 akan bergerak sampai posisi sudut motor 2 tercapai.

Ketelitian saat pengaturan koordinat pergerakan pada *flexible fixture* yang dalam hal ini sangat penting karena akan berpengaruh pada produk yang dihasilkan. Produk yang dihasilkan harus benar-benar baik dan sesuai dengan yang diharapkan. Oleh karena itu akurasi *flexible fixture* dan pemrogramannya sangat diperlukan pada proses ini.



Gambar 3. 3 Rangkaian Sistem Kontrol yang Direncanakan



Gambar 3. 4 Blok Diagram Sistem Kontrol *Flexible Fixture*

Gambar 3.3 merupakan rangkaian sistem kontrol yang direncanakan dan gambar 3.4 adalah blok diagram dari sistem kontrol *flexible fixture*, dimana set point (*potentiometer*) akan memberikan data kepada *microcontroller*. Kemudian *microcontroller* akan mengendalikan putaran *actuator* (*motor DC*), melalui perantara *switching device* (*motor DC driver*), karena tegangan output *microcontroller* adalah 5V. *Switching device* berperan sebagai *supply* listrik bagi *motor DC* yang mempunyai tegangan 24V. Sensor yang digunakan adalah *multi-turn potentiometer* yang berfungsi sebagai *feedback*.

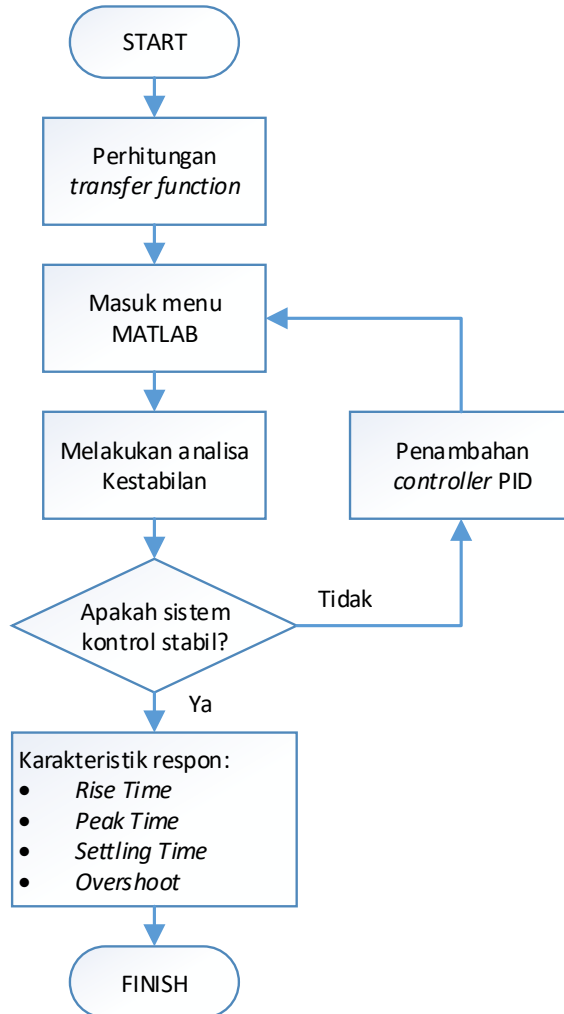
3.3 Analisa Sistem Kontrol *Flexible fixture* dengan *Microcontroller*

Pada tahap berikut ini melakukan analisa terhadap sistem kontrol yang berada di *flexible fixture*, yang hanya meliputi analisa tentang kepresisian sudut yang dicapai dari *motor DC*. Untuk tahap ini didapatkan informasi tentang tingkat akurasi sistem kontrol pada *motor DC*. Proses tersebut dilakukan dengan cara membandingkan sudut yang dicapai *motor DC* dengan busur derajat yang memiliki resolusi lebih tinggi dibandingkan dengan busur derajat yang ditempelkan pada *flexible fixture*.

3.4 Analisa Kestabilan Sistem Kontrol *Flexible Fixture*

Pada tahap berikut ini melakukan analisa terhadap sistem kontrol yang berada di *flexible fixture* yang hanya meliputi analisa terhadap kestabilan sistem kontrol. Untuk tahap ini didapatkan informasi tentang kestabilan sistem kontrol menggunakan bantuan MATLAB, jika untuk kenyataannya berguna agar sistem berjalan dengan baik dan stabil tanpa ada proses kontrol yang menyimpang dari program maupun kesalahan yang terjadi disaat proses permesinan berlangsung. Dengan *software* MATLAB dapat membantu menganalisa sistem kontrol dengan memberikan batasan-batasan yang diperlukan. Berikut ini adalah urutan proses cek kestabilan

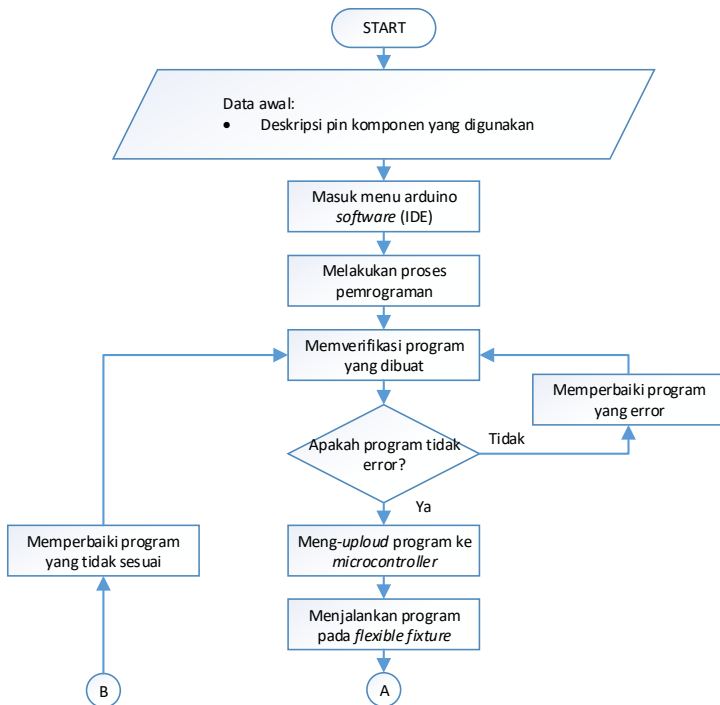
dengan *software* MATLAB yang ditunjukkan oleh gambar 3.5 dibawah ini.

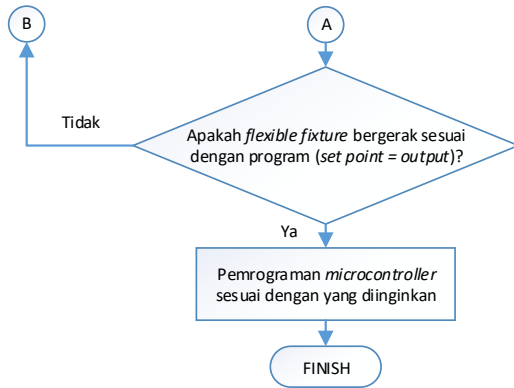


Gambar 3. 5 Diagram Simulasi MATLAB

3.5 Pemrograman Sistem Kontrol *Flexible Fixture*

Software yang digunakan untuk memprogram *microcontroller* adalah *arduino software (IDE)* yang dibuat dari bahasa pemrograman JAVA. Melalui *software* ini arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang diberikan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yaitu bahasa C. *Arduino Software (IDE)* ini dikembangkan dari *software processing* menjadi *arduino (IDE)* khusus untuk pemrograman dengan arduino. Gambar 3.6 dibawah ini merupakan diagram alir dalam pemrograman sistem kontrol *flexible fixture* pada *microcontroller*.





Gambar 3. 6 Pemrograman Sistem Kontrol *Flexible Fixture* Pada *Microcontroller*

BAB IV

PEMILIHAN KOMPONEN DAN ANALISA SISTEM KONTROL *FLEXIBLE FIXTURE*

4.1 Perencanaan Sistem Kontrol *Flexible fixture* Berbasis *Microcontroller*

Dalam perencanaan ini terdapat beberapa rangkaian yang menyusun suatu sistem kontrol yang berbasis *microcontroller*, antara lain:

- *Power supply*
- *Microcontroller*
- *Motor DC driver*
- *Motor DC*
- *Multi-turn potentiometer*

Dalam menghasilkan perputaran *motor DC* untuk mengontrol pergerakan *flexible fixture* diperlukan keempat rangkaian diatas. Salah satu proses yang terpenting adalah pemrograman pada komponen *microcontroller* dengan menggunakan *software* arduino (IDE).

4.1.1 *Power Supply*

Arus listrik yang digunakan di rumah, kantor dan pabrik pada umumnya adalah dibangkitkan, dikirim dan didistribusikan ke tempat masing-masing dalam bentuk arus bolak-balik atau arus AC (*Alternating Current*). Hal ini dikarenakan pembangkitan dan pendistribusian arus listrik melalui bentuk arus bolak-balik (AC) merupakan cara yang paling ekonomis dibandingkan dalam bentuk arus searah atau arus DC (*Direct Current*). Akan tetapi, peralatan elektronika yang digunakan sekarang ini sebagian besar membutuhkan arus DC dengan tegangan yang lebih rendah untuk pengoperasiannya. Oleh karena itu, peralatan elektronika harus memiliki sebuah rangkaian yang berfungsi untuk melakukan konversi arus listrik dari arus AC menjadi arus DC dan juga untuk menyediakan

tegangan yang sesuai dengan rangkaian elektronika, seperti diilustrasikan pada gambar 4.1 dibawah. Jika tegangan yang diberikan terlalu besar, maka peralatan elektronik akan cepat rusak, apabila tegangan terlalu kecil, maka peralatan elektronik tidak dapat bekerja dengan baik. Tegangan yang dibutuhkan pada peralatan elektronik kali ini adalah 24 V.

Transformator berfungsi untuk menurunkan tegangan listrik sesuai dengan kebutuhan komponen elektronika yang terdapat pada rangkaian adaptor (*DC power supply*). *Transformator* bekerja berdasarkan prinsip induksi elektromagnetik yang terdiri dari 2 bagian utama yang berbentuk lilitan yaitu lilitan primer dan lilitan sekunder. Lilitan primer merupakan input dari pada *transformator* sedangkan outputnya adalah pada lilitan sekunder.

Tegangan yang diturunkan oleh *transformator* masih berbentuk arus AC, sehingga harus diproses menggunakan *rectifier*. Rangkaian ini berfungsi untuk mengubah gelombang AC menjadi gelombang DC. *Rectifier* terdiri dari komponen dioda yang berjumlah 4 buah. Gelombang searah yang dihasilkan oleh dioda akan diteruskan pada kondensator yang akan menyimpan muatan listrik untuk sementara waktu, dan juga kondensator mempunyai fungsi lain yaitu sebagai *filter* dari *rectifier*, sehingga akan menghasilkan tegangan searah yang rata atau halus.



Gambar 4. 1 *Switching Power Supply*
(www.circuitspecialists.com)

4.1.2 *Microcontroller*

Sebagai otak pengendali dari gerakan *motor DC*, *microcontroller* ini sendiri tidak dapat langsung mengontrol dan memberi perintah pada *motor DC*, karena arus output *microcontroller* hanya 20 mA, sedangkan *motor DC* membutuhkan arus 5300 mA (5.3A) dan 560 mA (0.56A) untuk dapat berputar secara optimal. *Microcontroller* yang digunakan pada perencanaan ini adalah arduino MEGA (gambar 4.2), karena memiliki banyak keunggulan dibandingkan beberapa produk *microcontroller* lainnya, seperti mudahnya untuk diimplementasikan dengan perangkat lain baik dalam hal *hardware* atau *software* yang *open source*.



Gambar 4. 2 *Microcontroller* Arduino MEGA
(www.arduino.cc)

4.1.3 *Motor DC Driver*

Komponen utama dari rangkaian ini adalah transistor yang berjumlah 4 buah. Transistor tersebut berfungsi sebagai saklar (switching), karena arus yang dihasilkan yang dihasilkan oleh *microcontroller* kecil (20 mA), sehingga tidak dapat menggerakkan *motor DC* yang memerlukan arus 5300 mA (5.3A) dan 560 mA (0.56A). Pada perancangan kali ini driver yang digunakan adalah driver shield L293D. Ketika *microcontroller* memberikan logic 1 (5V) pada kaki *basis* dua transistor, maka arus dari *power supply* akan mengalir, hal ini menyebabkan motor

berputar atas perintah *microcontroller*. Rangkaian tersebut ditunjukkan oleh gambar 4.3 dibawah ini.



Gambar 4. 3 *Motor DC Driver*
(<https://playground.arduino.cc>)

4.1.4 *Motor DC*

Motor DC bertindak sebagai aktuator dalam perancangan ini. Terdapat 2 *motor DC* yang digunakan dengan spesifikasi yang berbeda. *Motor DC* 1 digunakan untuk memutar *chuck flexible fixture* terhadap sumbu z sejauh $0^{\circ} - 360^{\circ}$. *Motor DC* 2 digunakan untuk memutar *flexible fixtutre* terhadap sumbu y sejauh $0^{\circ} - 90^{\circ}$. Salah satu *motor DC* ditunjukkan pada gambar 4.4 dibawah ini.



Gambar 4. 4 *Motor DC*
(www.aliexpress.com)

4.1.4.1 Perhitungan Torsi dan Daya Motor

Perhitungan torsi dan daya motor berfungsi untuk mengetahui daya minimal motor agar dapat menggerakan

flexible fixture, perhitungan tersebut dapat dilihat dengan lengkap pada lampiran 1.

4.1.4.2 Spesifikasi Motor DC 1

Motor yang digunakan adalah motor DC dengan jenis DC022C-3 bertegangan input 24 V dan berdaya 14 W yang berfungsi untuk memutar *chuck flexible fixture* terhadap sumbu z sejauh $0^0 - 360^0$, selengkapnya dapat dilihat pada lampiran 2.

4.1.4.3 Spesifikasi Motor DC 2

Motor yang digunakan adalah motor DC dengan jenis DC048B-7 bertegangan input 24 V dan berdaya 127 W yang berfungsi untuk memiringkan *flexible fixture* terhadap sumbu y sejauh $0^0 - 90^0$, selengkapnya dapat dilihat pada lampiran 3.

4.1.5 *Multi-turn Potentiometer*

Multi-turn potentiometer berfungsi sebagai input dan sebagai feedback pergerakan dari *flexible fixture*. Nilai resistansi *Multi-turn potentiometer* adalah 100K Ohm. Alat ini beroperasi pada tegangan 5 V. *Multi-turn potentiometer* tersebut ditunjukkan pada gambar 4.5 dibawah ini.



Gambar 4. 5 *Multi-Turn Potentiometer*
(www.digikey.com)

4.2 Analisa Kecermatan *Multi-turn Potentiometer*

Arduino MEGA mempunyai resolusi 10 bit, sehingga

$$\begin{aligned} N_p &= 2^{10} \\ &= 1024 \end{aligned}$$

maka arduino MEGA dapat membaca nilai analog dari 0 sampai 1023. *Multi-turn potentiometer* memiliki 10 putaran, sehingga kecermatan sensor tersebut dapat ditentukan dengan rumus

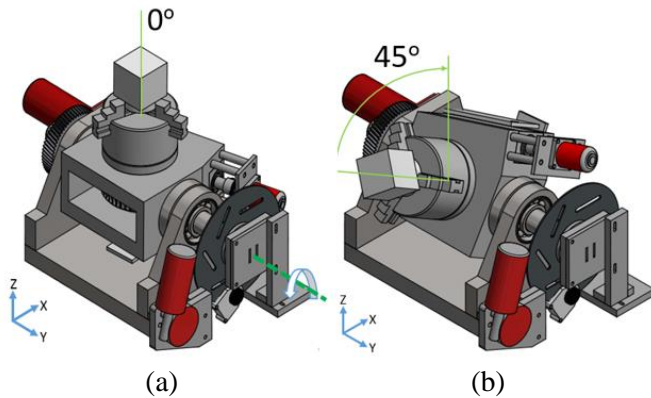
$$\begin{aligned} \delta &= \frac{10}{N_p} \cdot 360^\circ \\ \delta &= \frac{10}{1024} \cdot 360^\circ \\ \delta &= 3.51^\circ \end{aligned}$$

Berdasarkan perhitungan diatas, *multi-turn potentiometer* yang digunakan memiliki resolusi 3.51° pada setiap *step* atau pulsanya, sehingga menyebabkan adanya *error* pada sudut-sudut tertentu.

4.3 Analisa Pergerakan *Flexible fixture* Berbasis *Microcontroller*

4.3.1 Analisa Pergerakan Miring *Flexible Fixture*

Pada analisa pergerakan *flexible fixture* ini dilakukan dengan pengamatan langsung. *Flexible fixture* ini mampu memberikan kemiringan 90° terhadap sumbu y yang ditunjukkan oleh gambar 4.6.



Gambar 4. 6 Pergerakan Miring *Flexible Fixture*

Gambar 4.6(a) diatas merupakan salah satu contoh posisi awal *flexible fixture* dengan nilai set point sebesar 0^0 , dan gambar 4.6(b) merupakan salah satu contoh posisi akhir *flexible fixture* setelah dimasukkan nilai *set point* sebesar 45^0 .

Analisa pergerakan *flexible fixture* dapat dilakukan dengan menghitung persentase kesalahan dan persentase kepresisian sudut yang dihasilkan pada *flexible fixture* dengan rumus:

$$Kesalahan = \frac{\text{Sudut aktual} - \text{Sudut set point}}{\text{Sudut set point}} \times 100\%$$

$$Kepresisian = 100\% - \text{Kesalahan}(\%)$$

Dengan rumus diatas, maka didapatkan hasil analisa kesalahan dan kepresisian pada pergerakan miring *flexible fixture* yang ditunjukkan pada tabel 4.1:

Tabel 4. 1 Tabel Analisa Kesalahan dan Kepresisian
Pergerakan Miring *Flexible Fixture*

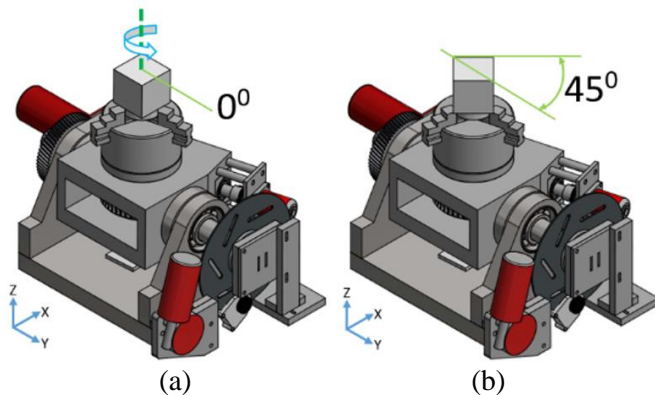
Sudut Setpoint (°)	Sudut Aktual (°)	Kesalahan (%)	Kepresisian (%)
5	4.58	8.40	91.60
10	10.25	2.50	97.50
15	14.42	3.87	96.13
20	19.67	1.65	98.35
25	24.75	1.00	99.00
30	30.33	1.10	98.90
35	34.92	0.23	99.77
40	40.33	0.82	99.18
45	44.83	0.38	99.62
50	50.42	0.84	99.16
55	55.33	0.60	99.40
60	59.67	0.55	99.45
65	64.67	0.51	99.49
70	70.42	0.60	99.40
75	75.25	0.33	99.67
80	79.92	0.10	99.90
85	85.33	0.39	99.61
90	90.42	0.47	99.53
Rata-Rata		1.35	98.65
Total		100	

Dari hasil tabel analisa kesalahan dan kepresisian pergerakan miring *flexible fixture* diatas dapat disimpulkan

bahwa rata-rata kesalahan yang terjadi adalah 1.35%, dan dengan tingkat kepresisian 98.65%. *Allowable error* yang digunakan adalah $\pm 2\%$, sehingga rata-rata kesalahan tersebut masih dalam batas toleransi, dan kesalahan tersebut dikarenakan adanya *backlash* yang terjadi pada pasangan *helix gear*.

4.3.2 Analisa Pergerakan Putar *Flexible Fixture*

Pada analisa pergerakan *flexible fixture* ini dilakukan dengan pengamatan langsung. *Flexible fixture* ini mampu berputar 360° terhadap sumbu z yang ditunjukkan oleh gambar 4.5 dibawah.



Gambar 4. 7 Pergerakan Putar *Flexible Fixture*

Gambar 4.7(a) diatas merupakan salah satu contoh posisi awal *flexible fixture* sebelum dimasukkan *set point* sebesar 0° , dan gambar 4.7(b) merupakan salah satu contoh posisi akhir *flexible fixture* setelah dimasukkan nilai *set point* sebesar 45° .

Analisa pergerakan *flexible fixture* dapat dilakukan dengan menghitung persentase kesalahan dan persentase kepresisian sudut yang dihasilkan pada *flexible fixture* dengan rumus:

$$Kesalahan = \frac{\text{Sudut aktual} - \text{Sudut set point}}{\text{Sudut set point}} \times 100\%$$

$$Kepresisian = 100\% - \text{Kesalahan}(\%)$$

Dengan rumus diatas, maka didapatkan hasil analisa kesalahan dan kepresisian pada pergerakan putar *flexible fixture* yang ditunjukkan pada tabel 4.2 dibawah ini:

Tabel 4. 2 Tabel Analisa Kesalahan Dan Kepresisian Pergerakan Putar *Flexible Fixture*

Sudut Setpoint (°)	Sudut Aktual (°)	Kesalahan (%)	Kepresisian (%)
5	5.42	8.40	91.60
10	10.33	3.30	96.70
15	14.67	2.20	97.80
20	20.08	0.40	99.60
25	25.25	1.00	99.00
30	30.33	1.10	98.90
35	35.17	0.49	99.51
40	40.42	1.05	98.95
45	44.75	0.56	99.44
50	50.33	0.66	99.34
55	54.83	0.31	99.69
60	60.33	0.55	99.45

Tabel 4. 2 Tabel Analisa Kesalahan dan Kepresisian
Pergerakan Putar *Flexible Fixture* (Lanjutan)

Sudut Setpoint (°)	Sudut Aktual (°)	Kesalahan (%)	Kepresisian (%)
65	64.67	0.51	99.49
70	69.67	0.47	99.53
75	75.42	0.56	99.44
80	80.17	0.21	99.79
85	85.08	0.09	99.91
90	89.58	0.47	99.53
95	94.83	0.18	99.82
100	100.42	0.42	99.58
105	105.25	0.24	99.76
110	109.75	0.23	99.77
115	114.75	0.22	99.78
120	119.58	0.35	99.65
125	125.25	0.20	99.80
130	130.17	0.13	99.87
135	134.67	0.24	99.76
140	139.67	0.24	99.76
145	144.83	0.12	99.88
150	150.33	0.22	99.78
155	155.42	0.27	99.73
160	159.75	0.16	99.84
165	165.25	0.15	99.85
170	170.42	0.25	99.75
175	174.92	0.05	99.95

Tabel 4. 2 Tabel Analisa Kesalahan dan Kepresisian
Pergerakan Putar *Flexible Fixture* (Lanjutan)

Sudut Setpoint (°)	Sudut Aktual (°)	Kesalahan (%)	Kepresisian (%)
180	179.92	0.04	99.96
185	185.33	0.18	99.82
190	190.42	0.22	99.78
195	195.42	0.22	99.78
200	199.67	0.17	99.84
205	205.33	0.16	99.84
210	210.42	0.20	99.80
215	215.42	0.20	99.80
220	220.08	0.04	99.96
225	224.58	0.19	99.81
230	229.67	0.14	99.86
235	234.92	0.03	99.97
240	240.25	0.10	99.90
245	245.42	0.17	99.83
250	250.33	0.13	99.87
255	255.17	0.07	99.93
260	260.25	0.10	99.90
265	264.83	0.06	99.94
270	269.92	0.03	99.97
275	275.33	0.12	99.88
280	279.83	0.06	99.94
285	284.83	0.06	99.94
290	290.25	0.09	99.91
295	295.17	0.06	99.94

Tabel 4. 2 Tabel Analisa Kesalahan dan Kepresisian Pergerakan Putar *Flexible Fixture* (Lanjutan)

Sudut Setpoint (°)	Sudut Aktual (°)	Kesalahan (%)	Kepresisian (%)
300	300.42	0.14	99.86
305	304.92	0.03	99.97
310	309.58	0.14	99.86
315	315.08	0.03	99.97
320	320.33	0.10	99.90
325	325.08	0.02	99.98
330	329.67	0.10	99.90
335	335.08	0.02	99.98
340	339.58	0.12	99.88
345	344.67	0.10	99.90
350	350.42	0.12	99.88
355	355.25	0.07	99.93
360	360.42	0.12	99.88
Rata-Rata		0.42	99.58
Total		100	

Dari hasil tabel analisa kesalahan dan kepresisian pergerakan miring *flexible fixture* diatas dapat disimpulkan bahwa rata-rata kesalahan yang terjadi adalah 0.42%, dan dengan tingkat kepresisian 99.58%. *Allowable error* yang digunakan adalah $\pm 2\%$, sehingga rata-rata kesalahan tersebut masih dalam batas toleransi, dan kesalahan tersebut dikarenakan adanya *backlash* yang terjadi pada pasangan *worm gear*.

4.4 Pembahasan Sistem Kontrol *Flexible Fixture*

Penelitian ini merupakan perancangan dan pembuatan sistem kontrol *flexible fixture* yang memiliki pergerakan 2 aksis yaitu pergerakan miring terhadap sumbu y sebesar 0^0 sampai dengan 90^0 (gambar 4.6), dan pergerakan putar terhadap sumbu z sejauh 0^0 sampai dengan 360^0 (gambar 4.7). Hasil akhir dari penelitian ini adalah *flexible fixture* yang dibuat dapat bergerak secara otomatis secara *continuous* sehingga tidak diperlukan tombol reset.

Cara kerja *flexible fixture* ini adalah dengan memberikan sinyal analog input terhadap *microcontroller* (arduino MEGA) yang telah dikonversi menjadi dalam satuan derajat dengan menggunakan komponen potensiometer, sehingga *controller* tersebut dapat menggerakkan *actuator* (motor DC) sesuai sinyal input yang dimasukkan, ini dapat terjadi karena pada *flexible fixture* dilengkapi dengan sensor *feedback* berupa potensiometer yang berfungsi sebagai umpan balik kepada *microcontroller* untuk menginformasikan apakah posisi sudut *flexible fixture* sudah sesuai dengan *set point* atau belum sesuai, jika sudut belum tercapai maka motor DC akan terus berputar *clockwise* maupun *counter clockwise* untuk mencapai sudut input yang diberikan (gambar 3.3 dan gambar 3.4).

Potensiometer yang digunakan dalam penelitian ini adalah jenis *multi-turn potentiometer* (gambar 4.5) yang memiliki 10 putaran dan spesifikasi *microcontroller* memiliki 10 bit yang bekerja pada tegangan 5 V. Hasil analisa menunjukkan tegangan tersebut apabila dibagi tiap sudut *flexible fixture* sebesar $0.00138 \text{ V}/^0$, dan kecermatan sensor yang sebesar 3.51^0 , sehingga mengakibatkan *flexible fixture* yang dibuat mempunyai kelemahan, yaitu *flexible fixture* hanya dapat bergerak sebesar sudut sesuai dengan kelipatan kecermatan sensor, dan motor DC tidak dapat berhenti pada sudut-sudut tertentu dikarena *microcontroller* terlambat dalam mengolah

input yang diberikan oleh sensor *feedback* dikarenakan RAM yang dimiliki Arduino MEGA sebesar 8 KB. Untuk penelitian selanjutnya dapat digunakan *multi-turn potentiometer* yang memiliki putaran yang lebih sedikit dan ditambahkan rangkaian *analog signal conditioning* yang berfungsi untuk meningkatkan kecermatan sensor *feedback*. Motor DC yang digunakan dalam pemodelan mempunyai spesifikasi yang berbeda dengan yang dianalisa, sehingga data-data motor yang diperlukan didapatkan dengan pendekatan dari motor DC bermerk berbeda.

“Halaman ini sengaja dikosongkan”

BAB V

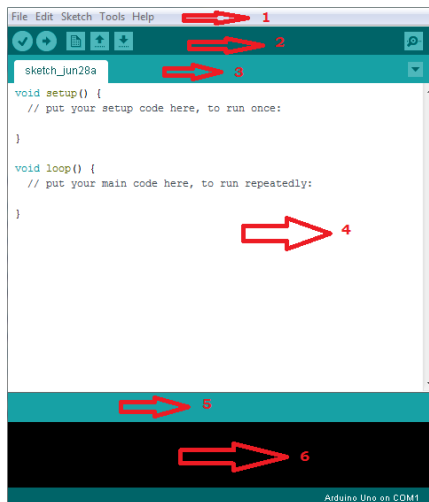
WIRING DIAGRAM DAN PEMROGRAMAN SISTEM KONTROL *FLEXIBLE FIXTURE*

5.1 Wiring Diagram Sistem Kontrol *Flexible Fixture*

Wiring diagram adalah gambaran suatu rangkaian listrik yang memberikan informasi secara detail, dari mulai simbol rangkaian sampai dengan koneksi rangkaian tersebut dengan komponen lain yang berfungsi untuk mempermudah mengikuti alur sebenarnya dari sebuah rangkaian sebagai peta dari sistem kelistrikan. Wiring diagram tersebut dapat dilihat selengkapnya pada lampiran 4.

5.2 Pemrograman Sistem Kontrol *Flexible Fixture* Pada *Microcontroller*

Pada tahap ini menggunakan *arduino software (IDE)* untuk pemrograman pada *microcontroller*. Tampilan *arduino software (IDE)* ditunjukkan gambar 5.1 dibawah ini.



Gambar 5. 1 Tampilan *Arduino Software (IDE)*

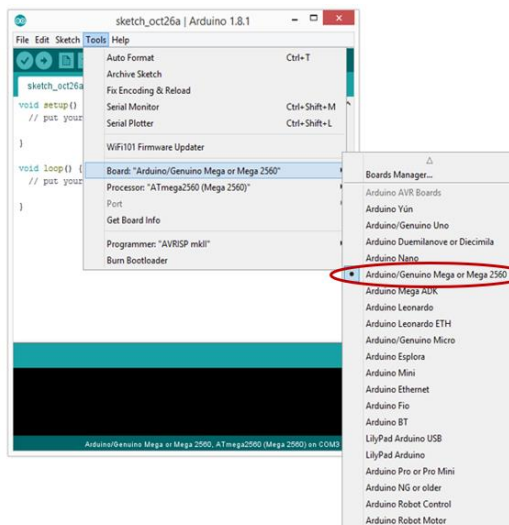
Berikut adalah penjelasan bagian-bagian arduino *software (IDE)*:

1. Menu arduino (*IDE*). Ada 5 menu yang membantu dalam melakukan pembuatan program, yaitu *file*, *edit*, *sketch*, *tools*, dan *helps*. Menu *file* digunakan untuk berinteraksi dengan *file *.ino*. Menu *edit* digunakan untuk mengedit program yang sedang ditulis di *sketch editor*. Menu *sketch* untuk mem-*verify*, meng-*upload sketch*. Menu *tools* untuk memanggil tools pendukung *software arduino (IDE)*. Menu *help* berisikan *file-file* dokumentasi yang berkaitan dengan masalah yang sering muncul serta penyelesaiannya. Selain itu pada menu *help* diberikan *link* untuk menuju Arduino Forum guna menanyakan serta mendiskusikan berbagai masalah yang ditemukan.
2. Kumpulan *shorcut* yang sering digunakan dalam proses pemrogram. *Verify* berfungsi untuk melakukan pengecekan kode yang dibuat apakah sudah sesuai dengan kaidah pemrograman. *Upload* berfungsi untuk melakukan kompilasi program atau kode yang dibuat menjadi bahasa yang dapat dipahami oleh arduino. *New* berfungsi untuk membuat *sketch* baru. *Open* berfungsi untuk membuka *sketch* yang pernah dibuat dan membuka kembali untuk dilakukan *editing* atau *upload* ulang ke arduino. *Save* berfungsi untuk menyimpan *sketch* yang telah kamu buat. *Serial Monitor* berfungsi untuk membuka jendela yang menampilkan data apa saja yang dikirimkan atau dipertukarkan antara arduino dengan *sketch* pada *port* serialnya. *Serial monitor* sangat berguna ketika ingin membuat program atau melakukan *debugging* tanpa menggunakan LCD pada arduino, yang dapat ditampilkan adalah nilai proses, nilai pembacaan, bahkan pesan error.
3. *Sketch Name*. Nama program atau *sketch*.

4. *Sketch Editor*. Tempat menulis program atau *sketch* arduino dalam bahasa C.
5. Jendela informasi atau notifikasi jika melakukan sesuatu pada *software* arduino (*IDE*).
6. Jendela *debug*. Jika ada kesalahan pembuatan program maka kesalahan tersebut akan muncul di jendela ini, sehingga akan memudahkan dalam mencari *bug* dalam program.

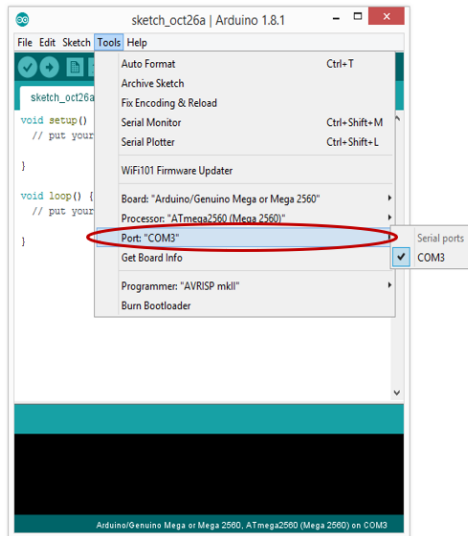
Berikut ini adalah urutan proses pemrograman pada *microcontroller* dengan arduino *software* (*IDE*).

- Hal pertama yang dilakukan adalah melakukan *setting board* arduino dengan memilih arduino yang digunakan. Pada penelitian ini dipilih *board* arduino mega atau mega 2560. Pemilihan tersebut ditunjukkan pada gambar 5.2 dibawah ini.



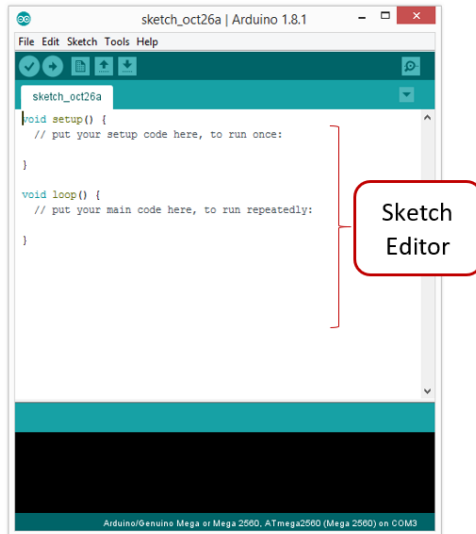
Gambar 5. 2 Tampilan Pemilihan *Board* Arduino Mega atau Pada Arduino *Software* (*IDE*)

- Melakukan pengecekan *software* arduino (*IDE*) dengan *microcontroller*, menu *port* akan aktif apabila *software* arduino (*IDE*) sudah terinstall dengan tepat. Pengecekan tersebut ditunjukkan pada gambar 5.3 dibawah ini.



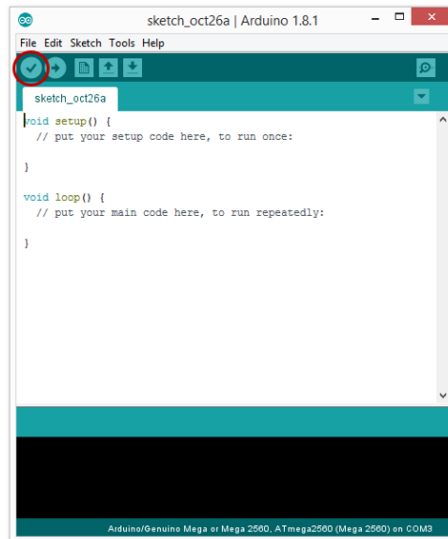
Gambar 5. 3 Tampilan Pengecekan Port Arduino Software (*IDE*) Dengan Microcontroller

- Melakukan pemrograman *microcontroller* pada lembar kerja atau *sketch editor* dengan menggunakan bahasa C. *Sketch editor* ditunjukkan pada gambar 5.4 dibawah ini.



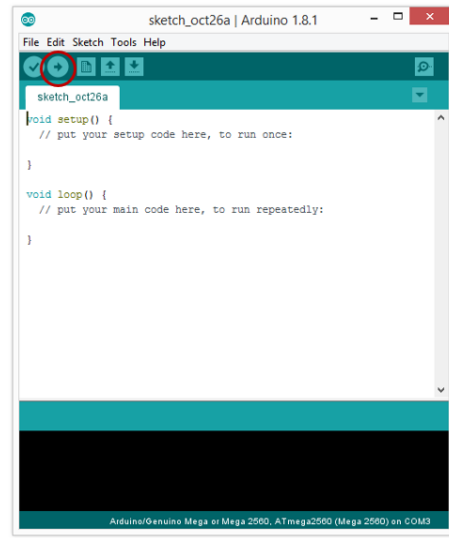
Gambar 5. 4 Tampilan *Sketch Editor* Pada *Arduino Software (IDE)*

- Setelah selesai proses pembuatan program langkah selanjutnya adalah mem-*verify* program, jika ada kesalahan pembuatan program maka kesalahan tersebut akan muncul di jendela *debug*. Menu *verify* ditunjukkan pada gambar 5.5 dibawah ini.



Gambar 5. 5 Tampilan Menu *Verify* Pada
Arduino Software (IDE)

- Langkah terakhir adalah meng-*upload* program ke *microcontroller*. Menu *upload* ditunjukkan pada gambar 5.6 dibawah ini.



Gambar 5. 6 Tampilan Menu *Upload* Pada *Arduino Software (Ide)*

5.3 *Sketch Program Sistem Kontrol Flexible Fixture*

Sketch program sistem kontrol *flexible fixture* dibuat pada lembar kerja atau *sketch editor* yang memiliki 3 bagian utama, yaitu deklarasi awal yang berfungsi untuk mendeskripsikan variabel-variabel yang akan digunakan dalam program utama dan juga untuk menambahkan file-file program yang dibutuhkan untuk menjalankan program utama, bagian *setup* yang digunakan untuk menginisiasi variabel, mengatur *mode pin* pada *board*, mengatur *timer*, dan sebagainya, bagian *setup* hanya dijalankan sekali saja yaitu pada saat awal program dijalankan atau ketika

program direset , dan program utama adalah fungsi utama program yang akan dijalankan berulang-ulang. *Sketch* program berisi perintah-perintah yang akan di-*upload* ke *microcontroller*, perintah tersebut ditulis dengan menggunakan bahasa C yang merupakan bahasa yang paling banyak digunakan dalam pemrograman *software* aplikasi. Setelah di-*upload*, perintah tersebut akan diolah oleh *microcontroller* dan *output*-nya akan diteruskan untuk mengatur putaran motor DC, melalui pasangan roda gigi, motor DC tersebut akan menggerakkan *flexible fixture* yang sesuai dengan perintah atau *set point* yang dikehendaki. *Sketch* program tersebut dapat dilihat selengkapnya dilampiran 5.

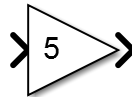
BAB VI

ANALISA KESTABILAN SISTEM KONTROL

6.1 Model Matematis Sistem Kontrol

➤ *Microcontroller*

Arduino MEGA adalah *microcontroller* yang beroperasi pada tegangan 5V, sehingga model matematis *microcontroller* mempunyai nilai *gain* 5. Model matematis atau fungsi alih tersebut ditunjukkan pada gambar 6.1 dibawah ini.

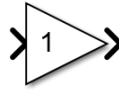


Microcontroller

Gambar 6. 1 Model Matematis Atau Fungsi Alih
Microcontroller

➤ *Motor DC Driver*

Pada penelitian ini, Motor DC *driver* merupakan komponen yang berfungsi untuk mengendalikan motor DC agar dapat bekerja. Cara kerja motor DC *driver* ini seperti halnya saklar *on-off* yang pada prinsipnya arus akan tersuplai ke motor DC apabila transistor dalam kondisi *ON*, dan arus akan terputus apabila dalam kondisi *OFF*. Gambar 6.2 dibawah ini merupakan ilustrasi model matematis atau fungsi alih motor DC *driver*.

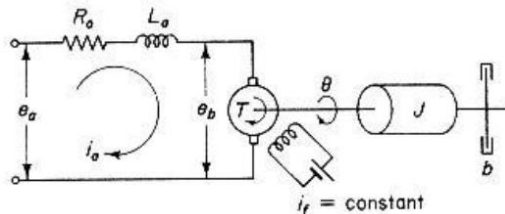


Motor DC Driver

Gambar 6. 2 Model Matematis Atau Fungsi Alih
Motor DC *Driver*

➤ Motor DC

Motor DC adalah rangkaian elektromekanik yang merupakan sistem gabungan antara elektrik dengan mekanik. Pada bagian elektrik terdapat resistansi dan induktansi, sedangkan bagian mekanik terdapat momen inersia, torsi, dan koefisien gesek. Diagram skematik motor DC ditunjukkan pada gambar 6.3 dibawah ini.



Gambar 6. 3 Diagram Skematik Motor DC

Dimana :

- R_a = tahanan kumparan jangkar (ohm)
- L_a = induktansi kumparan jangkar (henry)
- i_a = arus kumparan jangkar (ampere)
- i_f = arus medan (ampere)
- e_a = tegangan yang dikenakan pada jangkar (volt)
- e_b = ggl - gaya gerak listrik balik (volt)
- θ = perpindahan sudut dari poros motor (radian)

- T = torsi yang diberikan oleh motor (N.m)
- J = momen inersia ekivalen dari motor dan beban pada poros motor (kg.m²)
- b = koefisien gesekan viskos ekivalen dari motor dan beban pada poros motor (N.m.s/rad)

Motor DC memiliki data sebagai berikut:

Motor Data	Unit	Motor 1	Motor 2
J	kg-cm ²	0,0081	0,47
Ra	ohm	9	0,59
La	H	0,0047	0,00087
Kt=Kb	N-cm/A	2,63	7,06
B	N-cm-s/rad	0,00004	0,0017

Data-data motor DC diatas disubstitusikan kedalam persamaan (2.7) akan menjadi sebagai berikut:

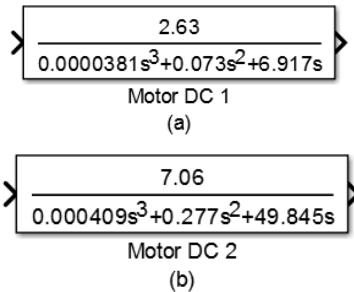
- Motor DC 1

$$\frac{\theta(s)}{E_a(s)} = \frac{2.63}{0.0000381s^3 + 0.073s^2 + 6.917s} \quad (6.1)$$

- Motor DC 2

$$\frac{\theta(s)}{E_a(s)} = \frac{7.06}{0.000409s^3 + 0.277s^2 + 49.845s} \quad (6.2)$$

Sehingga model matematis atau fungsi alih motor DC dalam *domain frequency* dapat ditunjukkan dalam gambar 6.4 dibawah ini.



Gambar 6. 4 (a) Model Matematis atau Fungsi Alih Motor DC 1,(b) Model Matematis atau Fungsi Alih Motor DC 2

➤ Gear

- Motor DC 1

Motor mempunyai *gear box* dengan ratio 4:1, dan *flexible fixture* mempunyai mekanisme *worm gear* dengan ratio 20:1, sehingga total *ratio* yang digunakan adalah 80:1.

Maka,

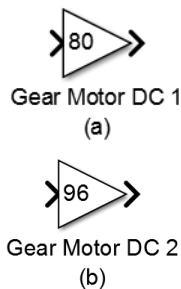
1 putaran poros motor = 1/80 putaran
flexible fixture

360° putaran poros motor = 4.5° putaran
flexible fixture

80° putaran poros motor = 1° putaran
flexible fixture

- Motor DC 2
 Motor mempunyai *gear box* dengan ratio 24:1, dan *flexible fixture* mempunyai mekanisme *helical gear* dengan ratio 4:1, sehingga total ratio yang digunakan adalah 96:1.
 Maka,
 1 putaran poros motor = 1/96 putaran *flexible fixture*
 360^0 putaran poros motor = 3.75^0 putaran *flexible fixture*
 96^0 putaran poros motor = 1^0 putaran *flexible fixture*

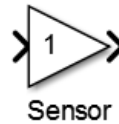
Sehingga dari analisa diatas, model matematis untuk gear dapat digambarkan pada gambar 6.5 dibawah ini.



Gambar 6. 5 (a) Model Matematis atau Fungsi Alih Gear Motor DC 1, (b) Model Matematis atau Fungsi Alih Gear Motor DC 2

➤ Sensor

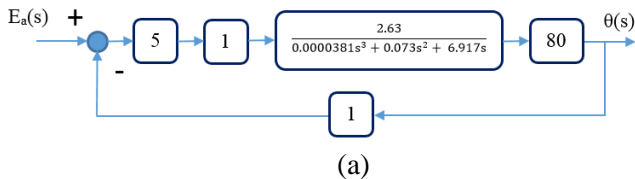
Sensor yang digunakan pada penelitian ini adalah *multi-turn potentiometer*, dimana cara kerjanya apabila pergerakan *flexible fixture* melebihi nilai *set point* maka akan memerintahkan motor DC untuk mundur, dan sebaliknya apabila pergerakan *flexible fixture* kurang dari *set point* maka akan memerintahkan motor DC untuk maju. Sehingga model matematis atau fungsi alih sensor digambarkan pada gambar 6.6 dibawah ini.

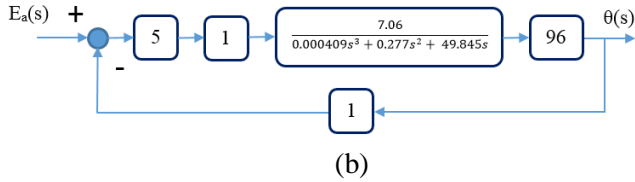


Gambar 6. 6 Model Matematis atau Fungsi Alih Sensor

6.2 Fungsi Transfer

Dari semua persamaan diatas dapat digambarkan model diagram *block* seperti yang ditunjukkan pada gambar 6.7 sebagai berikut,





Gambar 6. 7 (a) Diagram *Block* Sistem Kontrol *Flexible Fixture* Pada Motor DC 1, (b) Diagram *Block* Sistem Kontrol *Flexible Fixture* Pada Motor DC 2

Fungsi transfer dari diagram *block* sistem kontrol *flexible fixture* dengan analisa *negatif feedback* adalah sebagai berikut,

- Analisa *negatif feedback flexible fixture* pada motor DC 1

$$\frac{\theta(s)}{E_a(s)} = \frac{5 \cdot 1 \cdot \frac{2.63}{0.0000381s^3 + 0.073s^2 + 6.917s} \cdot 80}{1 + 5 \cdot 1 \cdot \frac{2.63}{0.0000381s^3 + 0.073s^2 + 6.917s} \cdot 80 \cdot 1} \quad (6.3)$$

- Analisa *negatif feedback flexible fixture* pada motor DC 2

$$\frac{\theta(s)}{E_a(s)} = \frac{5 \cdot 1 \cdot \frac{7.06}{0.000409s^3 + 0.277s^2 + 49.845s} \cdot 80}{1 + 5 \cdot 1 \cdot \frac{7.06}{0.000409s^3 + 0.277s^2 + 49.845s} \cdot 80 \cdot 1} \quad (6.4)$$

Setelah semua koefisien dimasukkan, maka didapatkan *overall transfer function* sebagai berikut,

- Sistem kontrol *flexible fixture* pada motor DC 1

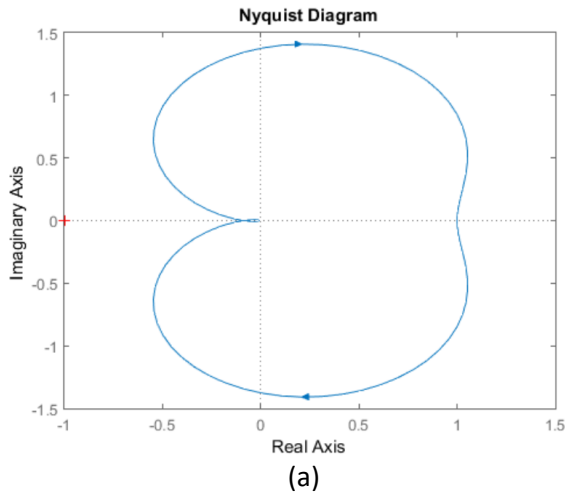
$$\frac{\theta(s)}{E_a(s)} = \frac{1052}{0.0000381s^3 + 0.073s^2 + 6.917s + 1052} \quad (6.5)$$

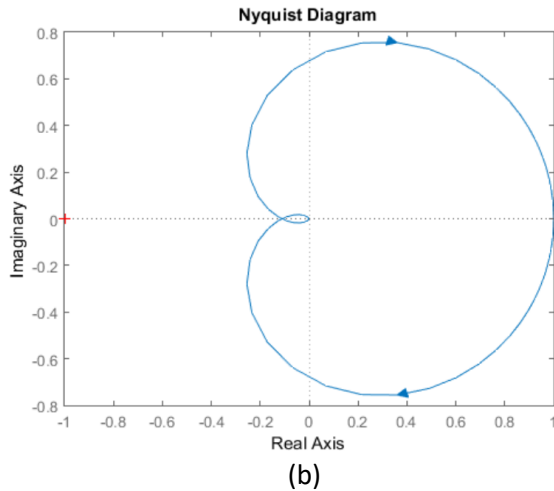
- Sistem kontrol *flexible fixture* pada motor DC 2

$$\frac{\theta(s)}{E_a(s)} = \frac{3389}{0.000409s^3 + 0.277s^2 + 49.845s + 3389} \quad (6.6)$$

6.3 Analisa Kestabilan dengan Metode Nyquist

Stabilitas dari sistem kontrol *loop* tertutup dapat ditentukan langsung dengan menghitung kutub dari fungsi transfer *loop* tertutup. Sebaliknya, kriteria kestabilan *nyquist* memungkinkan stabilitas ditentukan tanpa menghitung kutub *loop* tertutup. Diagram *nyquist* pada dasarnya *plot* dari $G(j\omega)$ dimana $G(s)$ adalah fungsi *loop* terbuka dan ω adalah vektor frekuensi yang berada dekat bidang sebelah kanan. Berikut gambar 6.8 menunjukkan hasil analisa kestabilan pada sistem kontrol *flexible fixture* dalam diagram *nyquist*





Gambar 6. 8 (a) *Nyquist Diagram Sistem Kontrol Flexible Fixture Pada Motor DC 1.* (b) *Nyquist Diagram Sistem Kontrol Flexible Fixture Motor DC 2*

Pada gambar 6.8 diatas terlihat bahwa *nyquist* diagram yang terbentuk tidak melingkupi titik $(-1, j0)$, sehingga dapat dikatakan bahwa sistem kontrol *flexible fixture* yang telah dibuat dalam keadaan stabil.

6.4 Analisa Kestabilan dengan Metode *Routh Hurwitz*

Kriteria kestabilan *routh hurwitz* merupakan kriteria yang menyatakan kestabilan absolut suatu sistem kontrol. Dimana pada kestabilan *routh hurwitz* menunjukkan ada atau tidaknya akar-akar dari persamaan karakteristik pada sistem kontrol yang terletak pada sisi sebelah kanan sumbu imajiner. Didapatkan hasil *overall transfer function* sistem sebagai berikut.

- Sistem kontrol *flexible fixture* pada motor DC 1

$$\frac{\theta(s)}{E_a(s)} = \frac{1052}{0.0000381s^3 + 0.073s^2 + 6.917s + 1052}$$

Dalam menganalisa kestabilan *routh hurwitz* ini, dari overall transfer function diatas dapat disusun koefisien-koefisien tabel *routh* dibawah ini.

	1	2
S^3	0,0000381	6,917
S^2	0,073	1052
S^1	a1	0
S^0	a0	

$$a1 = \frac{0.073 \cdot 6.917 - 0.0000381 \cdot 1052}{0.073}$$

$$a1 = 6.368$$

$$a0 = \frac{6.368 \cdot 1052 - 0.073 \cdot 0}{6.368}$$

$$a0 = 1052$$

Dari perhitungan diatas dapat dilihat bahwa hasil pada kolom pertama tidak terjadi perubahan tanda (semua positif), maka sistem kontrol *flexible fixture* dapat dikatakan stabil.

- Sistem kontrol *flexible fixture* pada motor DC 2

$$\frac{\theta(s)}{E_a(s)} = \frac{3389}{0.000409s^3 + 0.277s^2 + 49.845s + 3389}$$

Dalam menganalisa kestabilan *routh hurwitz* ini, dari *overall transfer function* diatas dapat

disusun koefisien-koefisien tabel *routh* dibawah ini.

	1	2
S^3	0,000409	49,845
S^2	0,277	3389
S^1	b1	0
S^0	b0	

$$b1 = \frac{0.277 \cdot 49.845 - 0.000409 \cdot 3389}{0.277}$$

$$b1 = 44.841$$

$$b0 = \frac{44.841 \cdot 3389 - 0.277 \cdot 0}{45.049}$$

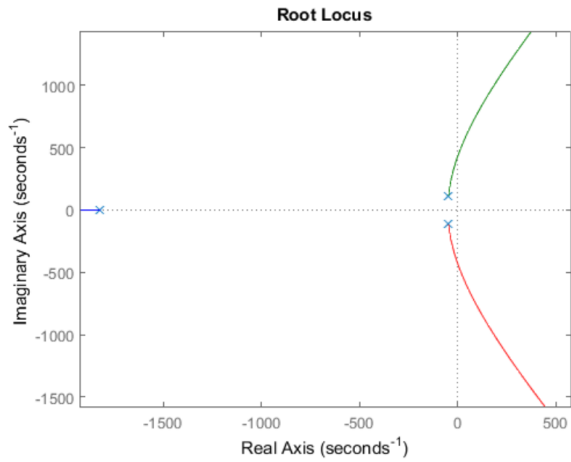
$$b0 = 3389$$

Dari perhitungan diatas dapat dilihat bahwa hasil pada kolom pertama tidak terjadi perubahan tanda (semua positif), maka sistem kontrol *flexible fixture* dapat dikatakan stabil.

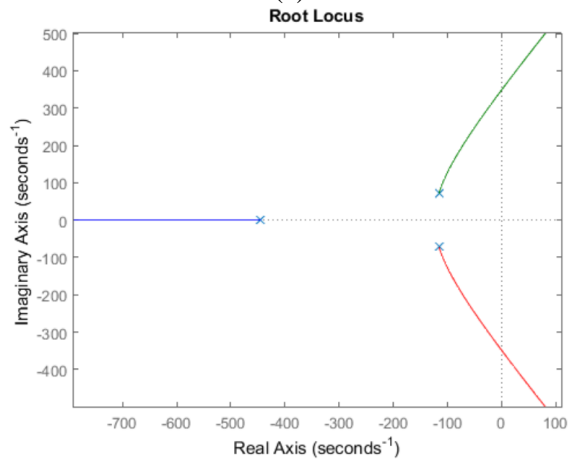
6.5 Analisa Kestabilan dengan Metode *Root Locus*

Root locus (tempat kedudukan akar) merupakan suatu analisis dalam *control engineering* yang menggambarkan pergeseran letak *pole-pole* suatu sistem *loop* tertutup dari perubahan besarnya penguatan. Analisis ini digunakan sebagai salah satu dasar untuk mendesain suatu sistem kendali sesuai dengan karakteristik dan spesifikasi yang diinginkan. Analisis *Root Locus* ini dapat menentukan apakah suatu system stabil atau tidak, selain itu dapat menentukan besarnya rentang penguatan *loop* tertutup agar suatu system masih dapat dikatakan stabil. Berikut

adalah gambar 6.9 hasil dari analisa dengan metode *root locus*.



(a)



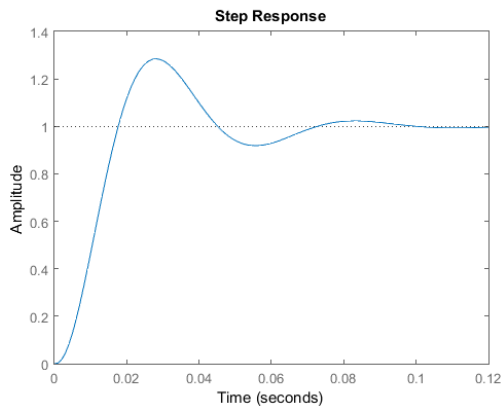
(b)

Gambar 6. 9 (a) *Root Locus* Sistem Kontrol *Flexible Fixture* Pada Motor DC 1, (b) *Root Locus* Sistem Kontrol *Flexible Fixture* Pada Motor DC 2

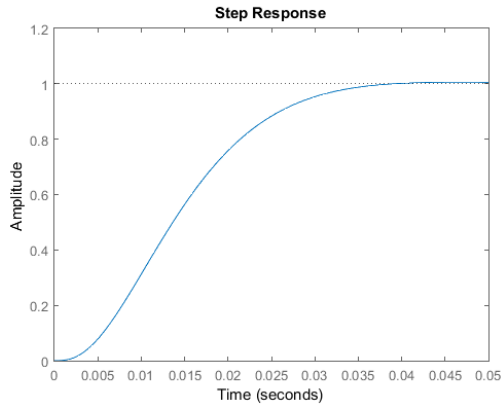
Dari hasil gambar 6.9 diatas dapat dikatakan bahwa sistem kontrol *flexible fixture* dalam keadaan stabil, karena akar-akar karakteristik dari *transfer function* berada pada sebelah kiri sumbu imajiner, yaitu -1824.8 , $-45.6 + 114.2i$, $-45.6 - 114.2i$ pada sistem kontrol *flexible fixture* pada motor DC 1, dan -445.41 , $-115.92 + 71.86i$, $-115.92 - 71.86i$ pada sistem kontrol *flexible fixture* motor DC 2.

6.6 Evaluasi Pergerakan Miring dan Putar Sistem Kontrol *Flexible Fixture*

Evaluasi sistem kontrol *flexible fixture* ini dengan memberi input *step response*, sehingga didapatkan nilai *peak time*, *settling time*, *rise time*, dan *percentage overshoot*. Evaluasi sistem kontrol ini berfungsi untuk memberi informasi mengenai kualitas dari respon sistem, yang artinya sistem kontrol dikatakan baik apabila sistem mencapai *steady state* dengan waktu yang cepat. Besaran-besaran tersebut dapat membantu perancang untuk membuat sistem kontrol mencapai *steady state* tanpa osilasi atau *overshoot* yang berlebihan dengan waktu yang diinginkan. Gambar 6.10 dibawah ini adalah hasil evaluasi sistem kontrol dengan input *step response*.



(a)



(b)

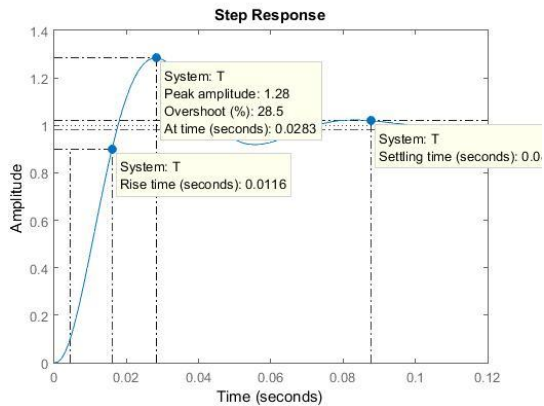
Gambar 6. 10 (a) Hasil Grafik *Step Response* Sistem Kontrol *Flexible Fixture* Pada Motor DC1, (b) Hasil Grafik *Step Response* Sistem Kontrol *Flexible Fixture* Pada Motor DC 2

Dari grafik *step response* diatas, dapat dilihat bahwa sistem kontrol pada motor DC 1 mengalami osilasi, tetapi setelah beberapa saat kemudian sistem akan mencapai *steady state*. Osilasi ini disebabkan karena sistem tidak mempunyai redaman atau damper, dan juga kontrol *flexible fixture* ini mempunyai *gain* yang besar. Pada sistem kontrol motor DC 2 mengalami osilasi yang lebih kecil, dan waktu untuk mencapai *steady state* lebih cepat sehingga dapat dikatakan sistem kontrol pada motor DC 2 lebih baik dibanding dengan sistem kontrol motor DC 1.

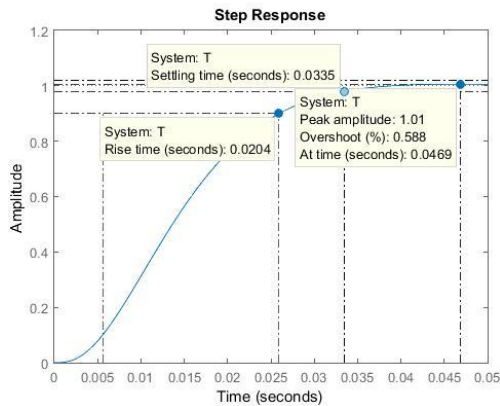
6.6.1 Karakteristik Respon

Dari gambar 6.10 diatas kita dapat mengetahui karakteristik respon yaitu *peak time* yang merupakan waktu yang diukur dari respon $t=0$ sampai mencapai puncak pertama, *settling time* yaitu waktu yang menyatakan respon telah

masuk dalam keadaan *steady state* dengan toleransi yang diestimasi sebesar $\pm 2\%$, *rise time* adalah waktu yang diukur dari respon $t=0$ sampai dengan memotong sumbu *steady state* pertama, dan *percentage overshoot* yang merupakan perbandingan antara nilai maksimum respon yang melampaui *steady state* dengan nilai *steady state*. Keempat karakteristik respon tersebut ditunjukkan pada gambar 6.11 dibawah ini.



(a)



(b)

Gambar 6. 11 (a) Karakteristik Respon Sistem Kontrol *Flexible Fixture* Motor DC 1, (b) Karakteristik Respon Sistem Kontrol *Flexible Fixture* Motor DC 2

Dari grafik *step response* diatas didapatkan nilai *time response* berupa *peak time* 0.0283 s, *settling time* 0.0877 s, *rise time* 0.0166 s, dan *persentase overshoot* 28.5% pada sistem kontrol *flexible fixture* motor DC 1, dan nilai *time response* berupa *peak time* 0.0469 s, *settling time* 0.0335 s, *rise time* 0.0204 s, dan *persentase overshoot* 0.588% pada sistem kontrol *flexible fixture* motor DC 2.

BAB VII KESIMPULAN DAN SARAN

7.1 Kesimpulan

Dari hasil rancangan yang telah dibuat, dapat diambil kesimpulan sebagai berikut:

1. Dari hasil tabel analisa kesalahan dan akurasi pergerakan miring *flexible fixture* dapat disimpulkan bahwa rata-rata kesalahan yang terjadi adalah 1.35%, dan dengan tingkat akurasi 98.65%. *Allowable error* yang digunakan adalah $\pm 2\%$, sehingga rata-rata kesalahan tersebut masih dalam batas toleransi, dan kesalahan tersebut dikarenakan adanya *backlash* yang terjadi pada pasangan *helix gear*. Dan hasil tabel analisa kesalahan dan akurasi pergerakan putar *flexible fixture* dapat disimpulkan bahwa rata-rata kesalahan yang terjadi adalah 0.42%, dan dengan tingkat akurasi 99.58%. *Allowable error* yang digunakan adalah $\pm 2\%$, sehingga rata-rata kesalahan tersebut masih dalam batas toleransi, dan kesalahan tersebut dikarenakan adanya *backlash* yang terjadi pada pasangan *worm gear*.

2. Sistem kontrol *flexible fixture* pada motor DC 1 mempunyai *overall transfer function*

$$\bullet \quad \frac{\theta(s)}{E_a(s)} = \frac{1052}{0.0000381s^3 + 0.073s^2 + 6.917s + 1052}$$

Sistem kontrol *flexible fixture* pada motor DC 2 mempunyai *overall transfer function*

$$\bullet \quad \frac{\theta(s)}{E_a(s)} = \frac{3389}{0.000409s^3 + 0.277s^2 + 49.845s + 3389}$$

3. Hasil analisa kestabilan kedua sistem kontrol dengan metode *nyquist* yaitu *nyquist* diagram yang terbentuk tidak melingkupi titik $(-1, j0)$, hasil analisa kestabilan pada kedua sistem kontrol dengan metode *routh*

hurwitz bahwa hasil pada kolom pertama tidak terjadi perubahan atau semua akar bernilai positif, dan hasil analisa kestabilan kedua sistem kontrol dengan metode *root locus* akar-akar karakteristik dari *transfer function* berada pada sebelah kiri sumbu imajiner, yaitu -1824.8 , $-45.6 + 114.2i$, $-45.6 - 114.2i$ pada sistem kontrol *flexible fixture* motor DC 1, dan -445.41 , $-115.92 + 71.86i$, $-115.92 - 71.86i$ pada sistem kontrol *flexible fixture* motor DC 2. Dari ketiga metode diatas dapat disimpulkan bahwa sistem kontrol *flexible fixture* dikatakan stabil.

4. Evaluasi sistem kontrol dengan memberi input *step response* didapatkan nilai *time response* berupa *peak time* 0.0283 s, *settling time* 0.0877 s, *rise time* 0.0166 s, dan *persentase overshoot* 28.5% pada sistem kontrol *flexible fixture* motor DC 1, dan nilai *time response* berupa *peak time* 0.0469 s, *settling time* 0.0335 s, *rise time* 0.0204 s, dan *persentase overshoot* 0.588% pada sistem kontrol *flexible fixture* motor DC 2.

7.2 Saran

Saran yang diperlukan untuk pengembangan perancangan selanjutnya adalah sebagai berikut:

1. Untuk mendapatkan kepresisian yang lebih tinggi, maka untuk penelitian selanjutnya dibutuhkan motor servo dengan spesifikasi yang sesuai dengan kebutuhan, dan penggunaan sistem transmisi *ball worm* untuk mereduksi *backlash* yang terjadi.

2. Pada penelitian ini sensor *feedback* sistem kontrol yang digunakan tidak tahan apabila terkena cairan, sehingga untuk perancangan selanjutnya dibutuhkan sensor *feedback* yang dapat digunakan ketika proses pemesinan menggunakan cairan *cooling*.
3. Kecepatan sistem kontrol dalam mengolah input relatif lambat dikarenakan *microcontroller* arduino MEGA mempunyai RAM dengan kapasitas 8 Kb, sehingga diperlukan PLC yang mempunyai kecepatan operasi yang lebih cepat.
4. Untuk meningkatkan kecermatan sensor, pada perancangan selanjutnya penggunaan *multi-turn potentiometer* dapat diganti dengan *single-turn potentiometer* agar dapat mereduksi *error* pada sudut-sudut tertentu.

“Halaman ini sengaja dikosongkan”

DAFTAR PUSTAKA

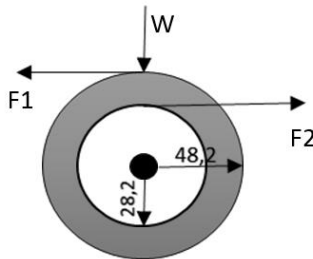
- [1] Johnson, C, D. 1993. **Procces Control Instrumentation Technology**. Jersey: Prentice Hall International Inc.
- [2] Nise, N, S. 2011. **Control Systems Engineering**. USA: John Wiley & Sons, Inc.
- [3] Bishop, R, H. 1998. **Modern Control System Analysis And Design Using MATLAB**. Texas: The University of Texas at Austin.
- [4] Ogata, K. 1995. **Teknik Kontrol Automatic**. Diterjemahkan oleh Edi Laksono. Jakarta: Erlangga.
- [5] Ala,P. 2011. **Perancangan Sistem Kontrol Flexible Fixture Berbasis Microcontroller**. Surabaya: ITS Surabaya.
- [6] Pratama, N. 2014. **Analisa Kontrol Posisi Motor Direct Current (DC) berbasis algoritma ANFIS Menggunakan Aplikasi Arduino**. Bandung: Universitas Telkom.
- [7] Arduino Board Mega. **Catalogue Arduino**. <URL:<https://www.arduino.cc/en/Main/ArduinoBoardMega>>.
- [8] Andika, D. 2014. **Motor DC**. <URL: http://www.academia.edu/9091244/MAKALAH_MOTOR_DC>.
- [9] Prasetyo, E. 2014. **Driver Motor DC H-Bridge**. <URL: <http://www.edukasiaelektronika.com/2014/07/driver-motor-dc-menggunakan-sistem-h.html>>.

“Halaman ini sengaja dikosongkan”

Lampiran 1. Perhitungan Torsi dan Daya Motor

- Motor 1

Dalam menghitung motor penggerak 1, gaya motor harus melebihi gaya gesek dari kedua bantalan yang digunakan.



Diketahui :

$$n = 50 \text{ rpm}$$

$$W = 500 \text{ N}$$

$$\mu = 0.04$$

Maka :

$$\begin{aligned} F2 &= 2\mu W \\ &= 40 \text{ N} \end{aligned}$$

$$T1 = T2$$

$$\begin{aligned} T2 &= F2 \times 28,2 \text{ mm} \\ &= 1128 \text{ Nm} \\ &= 1,128 \text{ Nm} \end{aligned}$$

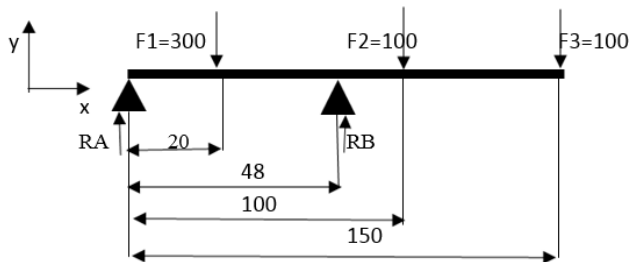
$$\begin{aligned} P &= \frac{n \times T2}{9550} \\ &= \frac{50 \times 1,128}{9550} \end{aligned}$$

$$= 0,0059 \text{ kW}$$

$$= 5,9 \text{ W}$$

Jadi 5.9 W adalah daya minimal agar *chuck flexible fixture* dapat berputar

- Motor 2



Diketahui :

$$n = 50 \text{ rpm}$$

$$F_{\text{total}} = 500 \text{ N}$$

Maka :

$$\sum M_A = 0$$

$$(F_1 \cdot 20) - (R_B \cdot 48) + (F_2 \cdot 100) + (F_3 \cdot 150) = 0$$

$$600 - 48R_B + 10000 + 15000 = 0$$

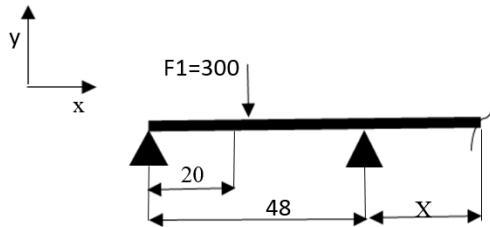
$$R_B = 645 \text{ N}$$

$$\sum F_y = 0$$

$$-R_A + F_1 + F_2 + F_3 - R_B = 0$$

$$-R_A = 300 + 100 + 100 - 645,83$$

$$R_A = 145,83 \text{ N}$$



$$\Sigma MB = 0$$

$$MB - RA(48 + x) - F_1(28 + x) + RB(x) = 0$$

Jika $x = 0$

$$MB - 145,83(48) - 300(28) = 0$$

$$MB = 15399,84 \text{ Nmm}$$

$$= 15,33 \text{ Nm}$$

$$\begin{aligned} P &= \frac{n \times T}{9550} \\ &= \frac{50 \cdot 15,33}{9550} \\ &= 0,080 \text{ kW} \\ &= 80 \text{ W} \end{aligned}$$

Jadi 80 W adalah daya minimal agar *flexible fixture* dapat bergerak.

“Halaman ini sengaja dikosongkan”

Lampiran 2. Spesifikasi motor DC 1

DC022C Series Salient Characteristics

MOTOR DATA	SYMBOL	UNITS	DC022C-1	DC022C-2	DC022C-3	OPTIONS
Max DC Terminal Voltage	V_r	V	36	36	36	Gearboxes G22A, G22B, PLG24
Max Speed (Mechanical)	ω_{MAX}	rpm	10000	10000	10000	
Continuous Stall Torque ¹	T_{cs}	Nm	0.0057	0.0093	0.014	
		oz-in	0.81	1.3	2.0	Encoders E22A
Peak Torque (Maximum) ¹	T_{pk}	Nm	0.018	0.037	0.066	
		oz-in	2.6	5.3	9.3	
Coulomb Friction Torque	T_f	Nm	9.2E-04	9.9E-04	0.0011	
		oz-in	0.13	0.14	0.15	
Viscous Damping Factor	D	Nm s/rad	2.7E-07	3.4E-07	4.0E-07	¹ Recorded at maximum winding temperature at 25°C ambient and without heatsink.
		oz-in/krpm	0.0040	0.0050	0.0060	
Thermal Time Constant	τ_{th}	min	9.9	11	13	
Thermal Resistance	R_{th}	°C/W	38	29	22	
Max. Winding Temperature	Θ_{MAX}	°C	130	130	130	
Rotor Inertia	J_r	kg m ²	5.2E-07	6.8E-07	8.1E-07	
		oz-in-s ²	7.3E-05	9.6E-05	1.1E-04	
Motor Weight	W_m	g	43	60	75	
		oz	1.5	2.1	2.7	

DC022C-3: Performance Data

MOTOR DATA	SYMBOL	UNITS								
Rated Voltage V1	V_r	V	6.00	7.58	9.55	12.0	15.2	19.1	24.0	30.3
Rated Torque ¹ ●	T_r	Nm	0.013	0.013	0.013	0.013	0.012	0.012	0.012	0.012
		oz-in	1.8	1.9	1.8	1.8	1.8	1.8	1.7	1.7
Rated Speed ¹	ω_r	rpm	5740	5500	5980	6160	6490	6620	6610	6670
Rated Current ¹	I_r	A	2.3	1.9	1.5	1.1	0.90	0.71	0.56	0.45
Rated Power ¹	P_r	W	7.7	7.5	8.1	8.2	8.5	8.6	8.5	8.6
No Load Speed	ω_{nl}	rpm	8500	8810	8780	8500	8600	8620	8550	8580
No Load Current	I_{nl}	A	0.22	0.18	0.15	0.11	0.087	0.069	0.055	0.044
Rated Voltage V2	V_r	V	4.78	6.00	7.58	9.55	12.0	15.2	19.1	24.0
Rated Torque ¹ ●	T_r	Nm	0.013	0.014	0.013	0.013	0.013	0.013	0.013	0.013
		oz-in	1.9	1.9	1.9	1.9	1.9	1.8	1.8	1.8
Rated Speed ¹	ω_r	rpm	3730	3350	3870	4140	4380	4570	4580	4600
Rated Current ¹	I_r	A	2.4	2.0	1.6	1.2	0.93	0.74	0.58	0.46
Rated Power ¹	P_r	W	5.2	4.8	5.4	5.7	6.0	6.2	6.2	6.2
No Load Speed	ω_{nl}	rpm	6740	6930	6930	6730	6760	6830	6780	6770
No Load Current	I_{nl}	A	0.21	0.17	0.14	0.11	0.082	0.066	0.052	0.041
Motor Constant	K_M	Nm/v/W	0.0078	0.0073	0.0077	0.0083	0.0085	0.0086	0.0088	0.0088
		oz-in/v/W	1.1	1.0	1.1	1.2	1.2	1.2	1.2	1.2
Torque Constant	K_T	Nm/A	0.00657	0.00798	0.0101	0.0132	0.0165	0.0207	0.0263	0.0330
		oz-in/A	0.930	1.13	1.43	1.87	2.34	2.93	3.72	4.68
Voltage Constant	K_E	V s/rad	0.00657	0.00798	0.0101	0.0132	0.0165	0.0207	0.0263	0.0330
		V/krpm	0.688	0.836	1.06	1.38	1.73	2.17	2.75	3.46
Terminal Resistance	R_{mt}	Ω	0.710	1.21	1.72	2.55	3.76	5.73	9.00	14.1
Inductance	L	mH	0.29	0.44	0.71	1.2	1.9	2.9	4.7	7.5
Peak Current	I_{pk}	A	8.5	6.3	5.6	4.7	4.0	3.3	2.7	2.1
Electrical Time Constant	τ_e	ms	0.41	0.36	0.41	0.46	0.50	0.51	0.52	0.53
Mechanical Time Constant	τ_m	ms	13	15	14	12	11	11	10	10

“Halaman ini sengaja dikosongkan”

Lampiran 3. Spesifikasi motor DC 2

DC054B Series Salient Characteristics

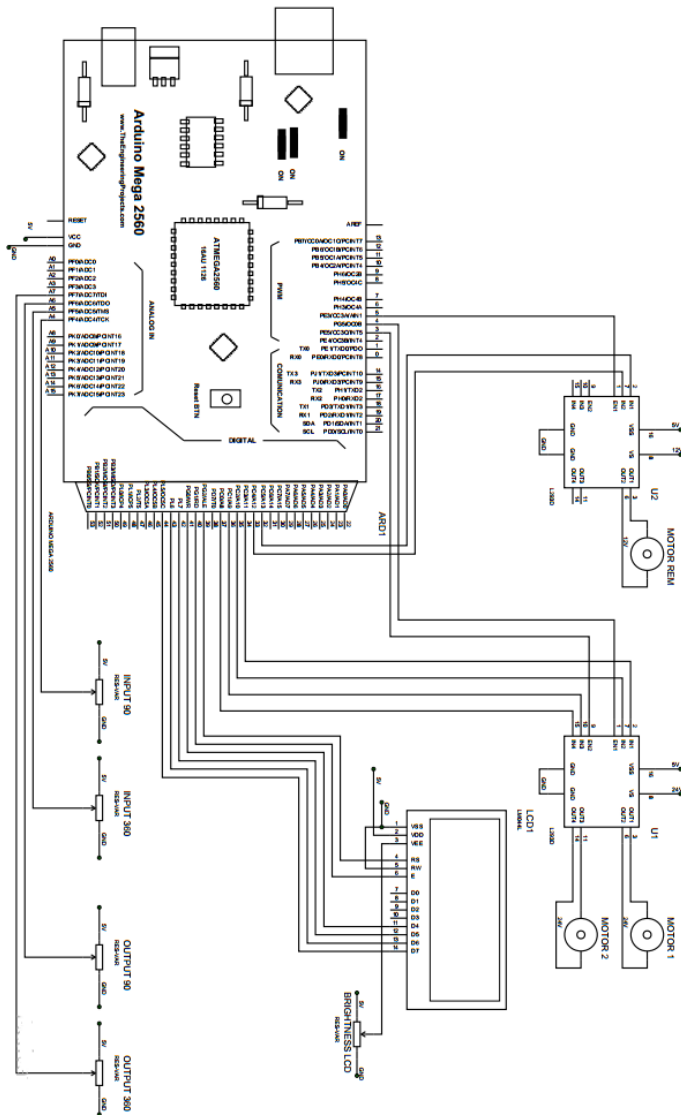
MOTOR DATA	SYMBOL	UNITS	DC 054B-1	DC 054B-2	DC 054B-3	DC 054B-4	DC 054B-5	DC 054B-6	DC 054B-7
Max DC Terminal Voltage	V_T	V	80	80	80	80	80	80	80
Max Speed (Mechanical)	ω_{MAX}	rpm	6000	6000	5000	5000	5000	5000	5000
Continuous Stall Torque ¹	T_{CS}	Nm	0.071	0.099	0.15	0.18	0.22	0.26	0.35
		oz-in	10	14	21	26	31	37	50
Peak Torque (Maximum) ¹	T_{pk}	Nm	0.39	0.67	1.0	1.3	1.4	1.8	2.6
		oz-in	55	95	140	180	200	260	370
Coulomb Friction Torque	T_f	Nm	0.0085	0.0085	0.011	0.011	0.014	0.014	0.016
		oz-in	1.2	1.2	1.6	1.6	2.0	2.0	2.2
Viscous Damping Factor	D	Nm s/rad	1.1E-05	1.1E-05	1.2E-05	1.2E-05	1.3E-05	1.3E-05	1.7E-05
		oz-in/krpm	0.17	0.17	0.18	0.18	0.19	0.19	0.25
Thermal Time Constant	τ_{th}	min	22	24	26	29	29	34	32
Thermal Resistance	R_{th}	°C/W	9.9	9.0	8.1	7.7	7.3	6.8	5.0
Max. Winding Temperature	Θ_{MAX}	°C	155	155	155	155	155	155	155
Rotor Inertia	J_r	kg m ²	1.1E-05	1.6E-05	2.1E-05	2.6E-05	3.1E-05	3.7E-05	4.7E-05
		oz-in-s ²	0.0016	0.0023	0.0030	0.0037	0.0044	0.0052	0.0067
Motor Weight	W_m	g	590	740	880	1000	1100	1300	1500
		oz	21	26	31	35	40	45	55

DC054B-7: Performance Data

MOTOR DATA	SYMBOL	UNITS								
Rated Voltage V1	V_r	V	15.2	19.1	24.0	30.3	38.2	48.0	60.6	76.4
Rated Torque ¹ ●	T_r	Nm	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31
		oz-in	44	44	44	44	44	44	43	43
Rated Speed ¹	ω_r	rpm	3000	2850	2900	2900	2920	2930	2930	2920
Rated Current ¹	I_r	A	8.9	6.7	5.3	4.2	3.3	2.6	2.1	1.6
Rated Power ¹	P_r	W	98	93	94	94	94	94	94	94
No Load Speed	ω_{nl}	rpm	3330	3140	3160	3140	3150	3160	3150	3140
No Load Current	I_{nl}	A	0.51	0.38	0.30	0.24	0.19	0.15	0.12	0.094
Rated Voltage V2	V_r	V	19.1	24.0	30.3	38.2	48.0	60.6	76.4	60.6
Rated Torque ¹ ●	T_r	Nm	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.32
		oz-in	41	41	41	41	41	40	40	46
Rated Speed ¹	ω_r	rpm	4020	3820	3890	3880	3880	3920	3910	2150
Rated Current ¹	I_r	A	8.4	6.3	5.0	3.9	3.1	2.5	2.0	1.7
Rated Power ¹	P_r	W	120	120	120	120	120	120	120	72
No Load Speed	ω_{nl}	rpm	4190	3950	3990	3960	3960	3990	3980	2480
No Load Current	I_{nl}	A	0.55	0.40	0.33	0.26	0.20	0.16	0.13	0.089
Motor Constant	K_M	Nm/v/W	0.086	0.091	0.092	0.093	0.094	0.093	0.094	0.094
		oz-in/v/W	12	13	13	13	13	13	13	13
Torque Constant	KV_T	Nm/A	0.0424	0.0565	0.0706	0.0897	0.113	0.141	0.179	0.226
		oz-in/A	6.00	8.01	9.99	12.7	16.0	20.0	25.3	32.0
Voltage Constant	K_E	V s/rad	0.0424	0.0565	0.0706	0.0897	0.113	0.141	0.179	0.226
		V/krpm	4.44	5.92	7.39	9.39	11.8	14.8	18.7	23.7
Terminal Resistance	R_{mt}	Ω	0.240	0.390	0.590	0.930	1.46	2.29	3.64	5.78
Inductance	L	mH	0.31	0.56	0.87	1.4	2.2	3.5	5.6	8.9
Peak Current	I_{pk}	A	63	49	41	33	26	21	17	13
Electrical Time Constant	τ_e	ms	1.3	1.4	1.5	1.5	1.5	1.5	1.5	1.5
Mechanical Time Constant	τ_m	ms	6.3	5.8	5.6	5.5	5.4	5.4	5.4	5.4

“Halaman ini sengaja dikosongkan”

Lampiran 4. Wiring Diagram Sistem Kontrol *Flexible Fixture*



“Halaman ini sengaja dikosongkan”

Lampiran 5. Sketch Program Sistem Kontrol *Flexible Fixture*

```
#include <LiquidCrystal_I2C.h>
#include <AFMotor.h>
#include <Wire.h>
#include <Arduino.h>
```

```
LiquidCrystal_I2C lcd(0x3F,2,1,0,4,5,6,7,3,POSITIVE);
```

```
int Pin90      = A2;
int Pin360     = A1;
int Pout90     = A4;
int Pout360    = A0;
```

```
const int PinStart      = 36;
const int PinReset90    = 39;
const int ledPin        = 42;
```

```
int PosisiReset90      = 1000;
int PosisiReset360     = 200;
```

```
int setValuePin90      = 0;
int setValuePout90     = 0;
int setValuePin360     = 0;
int setValuePout360    = 0;
```

```
int outputValuePin90   = 0;
int outputValuePout90  = 0;
int outputValuePin360  = 0;
int outputValuePout360 = 0;
```

```
int ButtonStartState   =0;
int BtnStartState      =0;
int ButtonResetState90 =0;
int BtnResetState90;
```

```
int lastButtonState           = LOW;
int lastButtonresetState      = LOW;
unsigned long lastDebounceTime = 0;
unsigned long debounceDelay   = 50;
int kondisitombolstartState    = LOW;
int kondisitombolresetState    = LOW;
```

```
AF_DCMotor motor1(1, MOTOR12_64KHZ);
AF_DCMotor motor3(3, MOTOR12_64KHZ);
AF_DCMotor motor4(4, MOTOR12_64KHZ);
```

```
int toleransi90  = 0.25;
int toleransibawah90;
int toleransiatas90;
```

```
int toleransi360 = 0.25;
int toleransibawah360;
int toleransiatas360;
```

```
int d090      = 15150; //untuk kalibrasi sudut 90
int d9090      = 16080;
int d0360      = 15240; //untuk kalibrasi sudut 360
int d360360     = 13440;
```

```
int kondisi90      =0;
int kondisi360      =0;
int kondisilcd90    =0;
int kondisilcd360   =0;
```

```
int titiknolinput90;
int inputderajat90;
```

```
int titiknoloutput90;
int outputderajat90;
```

```
int titiknolininput360;  
int inputderajat360;
```

```
int titiknoloutput360;  
int outputderajat360;
```

```
int kondisirem;  
int kondisirem2;
```

```
#define maju 15  
#define mundur 16
```

```
void setup() {  
  Serial.begin(115200);  
  Serial.println("Controller Ready...");  
  lcd.begin (20,4);
```

```
  kondisi90      =0;  
  kondisi360     =0;  
  kondisilcd90   =0;  
  kondisilcd360  =0;  
  kondisirem     =0;  
  kondisirem2    =2;
```

```
  ButtonStartState    =0;  
  ButtonResetState90  =0;
```

```
  motor1.setSpeed(225); //kecepatan motor 90  
  motor3.setSpeed(225); //kecepatan motor 90  
  motor4.setSpeed(255); //kecepatan motor 360  
  pinMode(maju, OUTPUT);  
  pinMode(mundur, OUTPUT);
```

```
  pinMode(PinStart, INPUT);  
  pinMode(PinReset90, INPUT);
```

```

    pinMode(ledPin, OUTPUT);
}

void loop() {
    jalan();
    rem();
    delay(1000);
    lepasrem();
    delay(1000);
}

void rem(){
    digitalWrite(mundur,LOW);
    digitalWrite(maju,HIGH);
    Serial.println("rem");
    delay(1745);
    digitalWrite(mundur,LOW);
    digitalWrite(maju,LOW);
}

void lepasrem(){
    digitalWrite(maju,LOW);
    digitalWrite(mundur,HIGH);
    Serial.println("lepas rem");
    delay(2000);
    digitalWrite(mundur,LOW);
    digitalWrite(maju,LOW);
}

void jalan() {
    int ButtonStartState          = 0;
    int ButtonResetState90        = 0;

    long setValuePin90 = analogRead (Pin90);

```

```
long outputValuePin90 = map(setValuePin90, 0,1023,0,810);  
long setValuePout90 = analogRead (Pout90);  
long outputValuePout90 = map(setValuePout90,  
0,1023,0,30690);
```

```
int titiknolinput90=(outputValuePin90)+d090;  
int inputderajat90=(titiknolinput90-d090)/9;
```

```
int titiknoloutput90=outputValuePout90-d090;  
int outputderajat90=titiknoloutput90/9;
```

```
toleransibawah90=inputderajat90-toleransi90;  
toleransiatas90=inputderajat90+toleransi90;
```

```
long setValuePin360 = analogRead (Pin360);  
long outputValuePin360 = map(setValuePin360,  
0,1023,0,3069);  
long setValuePout360 = analogRead (Pout360);  
long outputValuePout360 = map(setValuePout360,  
0,1023,0,30690);
```

```
int titiknolinput360=(outputValuePin360)+d0360;  
int inputderajat360=(titiknolinput360-d0360)/8.525;
```

```
int titiknoloutput360=outputValuePout360-d0360;  
int outputderajat360=titiknoloutput360/8.525;
```

```
toleransibawah360=inputderajat360-toleransi360;  
toleransiatas360=inputderajat360+toleransi360;
```

```
lcd.clear();  
lcd.setCursor(0, 0); //baris pertama  
lcd.print("90 : I= ");  
lcd.setCursor(8, 0);  
lcd.print("    ");
```



```
lcd.setCursor(8, 0);  
lcd.print(inputderajat90);
```

```
lcd.setCursor(11, 0);  
lcd.print(" O= ");  
lcd.setCursor(14, 0);  
lcd.print("    ");  
lcd.setCursor(14, 0);  
if(kondisilcd90==1){  
    lcd.print(inputderajat90);  
}  
else{  
    lcd.print(outputderajat90);  
}
```

```
lcd.setCursor(0, 1); //baris kedua  
lcd.print("360 : I= ");  
lcd.setCursor(8, 1);  
lcd.print("    ");  
lcd.setCursor(8, 1);  
lcd.print(inputderajat360);
```

```
lcd.setCursor(11, 1);  
lcd.print(" O= ");  
lcd.setCursor(14, 1);  
lcd.print("    ");  
lcd.setCursor(14, 1);  
if(kondisilcd360==1){  
    lcd.print(inputderajat360);  
}  
else{  
    lcd.print(outputderajat360);  
}
```

```
Serial.print("nilai derajat input 90 = ");
```

```
Serial.print(inputderajat90);  
Serial.print(" -- ");  
Serial.print("nilai derajat output 90 = ");  
Serial.print(outputderajat90);  
Serial.print(" -- ");  
Serial.print("nilai analog output 90 = ");  
Serial.print(outputValuePout90);  
Serial.print(" -- ");  
Serial.print("derajat toleransi 90 bawah= ");  
Serial.print(toleransibawah90);  
Serial.print(" -- ");  
Serial.print("derajat toleransi 90 atas = ");  
Serial.println(toleransiatas90);
```

```
Serial.print("nilai derajat input 360 = ");  
Serial.print(inputderajat360);  
Serial.print(" -- ");  
Serial.print("nilai derajat output 360 = ");  
Serial.print(outputderajat360);  
Serial.print(" -- ");  
Serial.print("nilai analog output 360 = ");  
Serial.print(outputValuePout360);  
Serial.print(" -- ");  
Serial.print("derajat toleransi 360 bawah= ");  
Serial.print(toleransibawah360);  
Serial.print(" -- ");  
Serial.print("derajat toleransi 360 atas = ");  
Serial.println(toleransiatas360);
```

```
//Jika tombol start ditekan  
int ButtonStartState = digitalRead(PinStart);  
Serial.println(ButtonStartState);
```

```
int ButtonResetState90 = digitalRead(PinReset90);  
Serial.println(ButtonResetState90);
```

```

if(ButtonStartState == LOW) {
    Serial.print("status start LOW");
    Serial.print(" -- ");
    Serial.println(ButtonStartState);
    kondisi90=1;
    kondisilcd90=0;
    kondisilcd360=0;

    digitalWrite(maju,LOW);
    digitalWrite(mundur,HIGH);
    Serial.println("lepas rem");
    delay(2000);
    digitalWrite(mundur,LOW);
    digitalWrite(maju,LOW);
}

Serial.print("kondisi 90 = ");
Serial.println(kondisi90);

if(kondisi90==1){
    if(outputderajat90<inputderajat90){
        motor3.run(BACKWARD);
        Serial.println("motor 90 maju");
    }
    else if(outputderajat90>inputderajat90){
        motor3.run(FORWARD);
        Serial.println("motor 90 mundur");

        int ButtonResetState90 = digitalRead(PinReset90);
        Serial.println(ButtonResetState90);

        if(ButtonResetState90 == LOW) {
            Serial.print("status reset LOW");
            Serial.print(" -- ");

```

```

    Serial.println(ButtonResetState90);
    if(kondisi90==1){
        motor3.run(BACKWARD);
        Serial.println("motor 90 maju");
    }
}
}
else if((outputderajat90>toleransibawah90) &&
(outputderajat90<toleransiatas90)){
    Serial.println("Nilai 90 sama");
    motor3.run(RELEASE);
    kondisi90=2;
    kondisi360=1;
    Serial.println("motor 90 berhenti");
    kondisilcd90=1;
    digitalWrite(mundur,LOW);
    digitalWrite(maju,HIGH);
    Serial.println("rem");
    delay(1745);
    digitalWrite(mundur,LOW);
    digitalWrite(maju,LOW);
}
delay(250);
}
if(kondisi360==1 and kondisi90==2){
    if(outputderajat360<inputderajat360){
        motor4.run(FORWARD);
        Serial.println("motor 360 mundur");
        Serial.print("kondisi 360 = ");
        Serial.println(kondisi360);
    }
    else if(outputderajat360>inputderajat360){
        motor4.run(BACKWARD);
        Serial.println("motor 360 maju");
        Serial.print("kondisi 360 = ");

```

```
    Serial.println(kondisi360);
}
else if((outputderajat360<toleransiatas360) &&
(outputderajat360>toleransibawah360)){
    kondisi360=0;
    Serial.print("kondisi 360 = ");
    Serial.println(kondisi360);
    Serial.println("Nilai 360 sama");
    motor4.run(RELEASE);
    Serial.println("motor 360 berhenti");
    kondisi90=0;
    Serial.print("kondisi 90 = ");
    Serial.println(kondisi90);
    kondisilcd360=1;
}
delay(250);
}
delay(250);
}
```

BIODATA PENULIS



Havif Alvyan Ikhwanudin dilahirkan di Boyolali, 03 Juli 1993, merupakan anak yang terlahir dari orang tua terbaik bernama Sarjono dan Khayatun. Riwayat pendidikan penulis diawali di MI Krasak pada tahun 1998-2004. Penulis melanjutkan pendidikannya di SMP N 1 Teras pada tahun 2004-2008, kemudian melanjutkan pendidikan di SMK N 2 Surakarta pada tahun 2008-2011, setelah itu penulis melanjutkan jenjang diploma di Politeknik ATMI Surakarta pada tahun 2011-2014, Selanjutnya penulis melanjutkan pendidikan jenjang sarjana jurusan Teknik Mesin di Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Motto hidup penulis adalah **“Man For Others”** menjadikan penulis menjalani hidup lebih peka terhadap keadaan sekitar sebagai manusia socius. Untuk semua informasi dan masukan terkait tugas akhir ini dapat menghubungi penulis melalui email alvyanhavif@gmail.com.

“Halaman ini sengaja dikosongkan”